ICLEval: Evaluating In-Context Learning Ability of Large Language Models

Anonymous ACL submission

Abstract

In-Context Learning (ICL) is a crucial capability of LLMs as it enables them to understand 003 and reason across a series of interconnected inputs. However, existing evaluation frameworks primarily focus on language abilities and knowledge, often neglecting the evaluation of ICL ability. This limitation hampers our un-007 derstanding of how LLMs utilize context in complex problem-solving. In this study, we introduce the ICLEval benchmark to assess the ICL abilities of LLMs systematically. We evaluate two fundamental abilities: copying and learning. We also investigate the impact of model size, pretraining stage, and other factors 014 015 on ICL abilities. Our findings reveal that model size is not the sole determinant of ICL efficacy. 017 Surprisingly, we observe that ICL abilities, particularly copying, develop early in the pretraining process and stabilize afterward. Furthermore, we discover that the ICL abilities also be influenced by some other factors, such as distinguishing ability, inherent preferences, attention points capacity, and tokenizer.

1 Introduction

024

034

038

040

Large language models (LLMs) possess stronger language abilities and richer knowledge compared to traditional models, but their most important differentiating factor is their in-context learning (ICL) ability (Brown et al., 2020). ICL ability serves as a hallmark of LLMs and plays a crucial role in their strong generalization performance. It enables LLMs to quickly adapt to new tasks without altering their internal parameters, utilizing techniques like zero-shot and few-shot learning. Moreover, ICL is a pivotal component in tool learning (Qin et al., 2023b) and various agent strategies, such as chain-of-thought (Wei et al., 2022) and tree-ofthought (Yao et al., 2023).

Presently, evaluations of large language models primarily focus on language abilities (Reddy et al., 2019; Lai et al., 2017; Rajpurkar et al., 2018) and



Figure 1: The ICL abilities of different models

042

043

045

047

051

054

056

060

061

062

063

064

knowledge applications (Clark et al., 2018; Bisk et al., 2020; Zellers et al., 2019; Mihaylov et al., 2018). Some studies have explored more complex abilities in LLMs, such as mathematics (Yuan et al., 2023), coding, and human exam tests (Hendrycks et al., 2020; Huang et al., 2023). However, there is currently no dedicated benchmark specifically designed to assess ICL ability. We can only indirectly gauge ICL ability through certain language ability test tasks or reasoning tasks. To gain a better understanding of the sources and influencing factors of ICL ability, a comprehensive evaluation is needed.

In this study, we introduce the ICLEval benchmark to systematically evaluate the ICL abilities of LLMs. We categorize ICL ability into two distinct dimensions: copying and learning. We believe copying is the foundational attribute of ICL abilities, as well as learning is the reflection of models' generalization. We design two tasks for the evaluation of copying ability for unstructured and structured contexts. Also, we design ten tasks for the evaluation of learning ability about four aspects: format, order, statistic, and complex rules.

Table 1: The number of samples in our ICL tasks.

Task Name	Туре	Num.
Copying		
Unstructured Text	String completion	100
Structured Text	Dictionary search	190
Learning		
-	Format check	120
Format	Format cloning	100
	Format conversion	120
Orden	Order check	100
Order	Order adjustment	240
	Duplication check	300
C4-4:-4:	De-duplication	300
Statistics	Count & navigation	120
	Relation analysis	100
Complex Rules	Numbers' rules	250
Total		2,040

096

065

We conduct experiments focusing on two aspects: model size and pretraining token number. Firstly, we test a series of base-version LLMs ranging from 1.1B to 65B parameters. We observe that larger models often exhibit stronger ICL ability. However, models of smaller size also can be compared with larger ones in ICL scores. Secondly, we evaluate three models (Baichuan2-7B, TinyLlama, and Amber) with different pretraining tokens. Surprisingly, we find that most ICL abilities reach their peak in the early stages of pretraining, with minimal growth during subsequent training. Moreover, we find some other factors that influence the ICL abilities due to our experiments' results. We analyze four factors: the distinguishing ability, inherent preferences, attention points capacity, and tokenizer in Section 3.4.

We hope that our work will contribute to the further research on the in-context learning ability of language models.

2 Benchmark Construction

In-context learning (ICL) is an inherent capability of large language models (LLMs), enabling them to comprehend and reason through a series of interconnected inputs. However, directly evaluating ICL poses challenges as it can easily be influenced by language abilities and knowledge. In this study, we aim to create evaluation scenarios for ICL that mitigate potential confounding factors.

94 2.1 In-Context Learning Tasks and Data

ICL primarily hinges on two foundational skills: (1) the copy ability, which involves seeking and aggregating related information from in-context examples; and (2) the learning ability, which enables the model to discern patterns and rules within the aggregated information relevant to the task at hand. Therefore, we design several tasks for evaluating the copy and learning abilities in ICLEval. We will introduce the evaluation tasks of the copy and learning abilities in the following. The categories and statistics for each task are presented in Table 1, and more details about samples can see Appendix A.

099

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

2.1.1 Copying Ability

Copying is a fundamental ability of the ICL mechanism, allowing LLMs to gather supplementary information from contextual cues, thereby enhancing their effectiveness in performing specific tasks. We posit that the efficacy of LLMs in copy information may exhibit variability across diverse scenarios. To evaluate the models' adaptability and proficiency in information acquisition within diverse contextual complexities, ICLEval divides copying tasks into two categories: unstructured (natural language) and structured contexts:

Copying in unstructured context is a similar scenario as the pretraining stage. We mask the second half of a string which appears multiple times in the previous paragraphs, and require the models to predict it given the first half (named string completion). This gauges the models' proficiency in context-based inference and replication.

Copying in structured context typically encountered in downstream applications. We choose a "dictionary" format as a representation of structured data. This task challenges the models to deduce a value from a specified key from a set of key-value pairs (named dictionary search), testing their ability to seek and extract information from structured repositories efficiently.

2.1.2 Learning Ability

The learning ability is another fundamental ability of the ICL mechanism, allowing LLMs to extract mapping rules from in-context examples, thus enabling them to tackle different and unseen tasks based on examples or natural language descriptions, without requiring updates to model parameters. To focus solely on the learning ability and avoid the influence of language understanding capabilities across different models, ICLEval decomposes the evaluation of learning ability into four foundational aspects: format rules, statistical rules, order rules, and complex rules:

Learning format rules assesses whether LLMs

Unstructured Text	Structured Text	Order
Prompt: The Baptist Mission Union censured 521351caba13f591 Calista and their spouses, who returned from America after their studies.\nIn 1872, long after the death of 521351ca Label: ba13f591	Prompt: c5f3b8 : On August 15, a 5baa0d : Moreover, cloudy days 79bbc3 : In this period, the 71df7d : In winning the 340b02 : Nevertheless, it has 71df7d : Label: In winning the	Prompt: Input: pain, leading, drag, shoot, Output: shoot, drag, pain, leading Input: shopping, drama, care, start Output: Label: start, care, shopping, drama
Format	Statistics	Complex Rules
Prompt: Input: Index name age city I1 Elijah Morgan 36 Pittsburgh Output: Label: <person> <name>Elijah Morgan</name> <age>36</age> <city>Pittsburgh</city> </person>	Prompt: Input: left, up, up, down, down, up, left, up, down Output: Label: up: 4, down: 3, left: 2, right: 0 count navigation Prompt: Input: right, right, up, left, down, up, down, left, down Output: Label: down: 1, left: 0	Prompt: Input: [1, 0, 3] Output: [1, 0, 1, 2, 3] Input: [55, 55, 50] Output: [55, 54, 53, 52, 51, 50] Input: [2, 5, 1, 7] Output: [2, 3, 4, 5, 4, 3,,, 5, 6, 7] Input: [9, 4, 2] Output: [9, 8, 7, 6, 5, 4, 3, 2] Input: [6, 5, 5, 1] Output: Label: [6, 5, 4, 3, 2, 1]

Figure 2: The examples of six representative scenarios in ICLEval.

can learn formatting mappings from in-context ex-148 amples, a key skill for generating appropriately for-149 150 matted outputs across various applications. Specifically, we design a format classification task and two 151 format generation tasks to evaluate LLMs' abilities 152 in format recognition (named format check), adher-153 ence to specified formats (named format cloning), 154 and format conversation. 155

156

157

158

159

160

161

162

163

164

165

166

167

169

170

171

172

173

174

175

176

177

Learning order rules examines LLMs' ability to grasp the order of elements (as opposed to the sequence of examples) from in-context examples. This skill is crucial for tasks requiring reorganization of input elements, such as in translation and syntactic analysis scenarios. To this end, we formulate a classification task and a generation task aimed at evaluating LLMs' capabilities in determining whether inputs satisfy a specific order (named order check) and in executing order transformations (order adjustment).

Learning statistic rules evaluates LLMs' ability to extract, filter, summarize, and analyze relevant information from in-context examples. We consider four typical tasks including detecting or eliminating duplicates within the context(named duplication check and de-duplication), counting the number of elements in the context (named count & navigation), and generating the related information about a given node in relation graph (named relation analysis). These tasks collectively aim to measure LLMs' analytical capabilities and their application in processing and interpreting complex information structures.

178

179

180

181

182

183

184

185

186

187

188

190

191

192

193

194

195

196

197

199

200

201

202

203

204

Learning complex rules is the task of finding diversity and complex rules from given in-context examples. ICLEval designs the task "numbers' rules" which is learning a mapping given multigroups of numbers list pairs. We use the data from "list_functions" task in BIGBench, and there are 250 diversity mapping rules in it with different difficulties.

All these tasks we designed utilize n-shot incontext examples while relying less on language abilities, commonsense knowledge, or factual knowledge.

3 Experiments

In the experiments, we first list the models evaluated in our work. Then, we analyze the effects of model size and pretraining tokens. Later, we give the analysis of four other factors impacting in-context learning ability. Due to the space limit, the whole results are provided in Appendix C.2.

3.1 Settings

Metrics We have a total of 12 tasks with 2,040 testing samples. For almost all tasks such as string completion, dictionary search, and format conversion, we use exact match scores to evaluate the predictions with the labels. But for the format cloning task, we only evaluate the correctness of the format

	Сору	ing		Le	arning		
Model	Unstructured Text	Structured Text	Format	Order	Statistics	Complex Rules	Average
TinyLlama-1.1B	0.89	0.46	0.60	0.46	0.23	0.36	0.50
Phi-1.5	0.54	0.25	0.48	0.46	0.25	0.39	0.40
GPT-Neo-1.3B	0.54	0.48	0.48	0.50	0.20	0.38	0.43
Yi-6B	0.87	0.38	0.61	0.61	0.28	0.49	0.54
ChatGLM3-6B	0.90	0.70	0.60	0.54	0.32	0.43	0.58
Qwen-7B	0.89	0.45	0.58	0.62	0.42	0.48	0.57
Baichuan2-7B	0.87	0.44	0.63	0.53	0.34	0.49	0.55
LLaMA-7B	0.87	0.33	0.68	0.54	0.27	0.42	0.52
LLaMA2-7B	0.89	0.45	0.61	0.63	0.32	0.45	0.56
Mistral-7B-v0.1	0.88	0.54	0.67	0.78	0.46	0.56	<u>0.65</u>
Qwen-14B	0.87	0.48	0.68	0.79	0.54	0.55	<u>0.65</u>
Baichuan2-13B	0.80	0.23	0.75	0.66	0.40	0.39	0.54
LLaMA-13B	0.89	0.43	0.63	0.55	0.32	0.47	0.55
LLaMA2-13B	0.89	0.47	0.60	0.72	0.36	0.52	0.59
Yi-34B	0.86	0.60	0.63	0.84	0.59	0.54	0.68
LLaMA-30B	0.89	0.33	0.73	0.77	0.46	0.53	0.62
LLaMA-65B	0.89	0.49	0.59	0.85	0.47	0.58	<u>0.65</u>

Table 2: Performance of LLMs with different sizes and the last column is the mean score of all aspects.

and do not consider the content. Moreover, we use
postprocessing to convert models' responses for
different tasks. More processing details are shown
in Appendix A.3.

Evaluated Models We evaluate various open-210 sourced LLMs with different model sizes, such as the LLaMA series, the Baichuan series, and the 212 Qwen series. For example, we evaluate the LLaMA 213 series with 7B, 13B, 34B, and 65B versions as well 214 as base-version and chat-version. Moreover, we 215 test the intermediate checkpoints with different pre-216 training stages for TinyLlama-1.1B, Baichuan-7B, 217 and Amber-7B. The detailed description of these 218 models can be found in Appendix **B**. 219

Inference Settings For all the models, whether
base-version or chat-version, we don't use additional prompts (e.g. "User:"), and use n-shot examples in most tasks. We don't use sampling or beam
search and only use the greedy decoding method.

226

229

234

3.2 How do Model Sizes Effect ICL Ability?

We divided the models into three groups: smallsized models (around 1B parameters), middle-sized models (around 10B parameters), and large-sized models (bigger than 30B parameters), and conducted tests on all these models. The results are presented in Table 2. From the table, we can obtain the following research findings:

(1) A general trend indicates that larger models
 tend to exhibit superior ICL performance. However, the data also presents a considerable variance within models of similar sizes. Notably, some

middle-sized models (e.g. such as Mistral-7B-v0.1 and Qwen-14B) demonstrate strong ICL abilities comparable to models having 5x-10x size (e.g. LLaMA-65B). This finding underscores that model size is not the sole determinant of ICL efficacy.

237

239

240

241

242

243

244

245

246

247

248

249

250

251

254

255

256

257

258

259

260

261

262

263

264

265

(2) For the copying ability, larger models don't have obvious advantages. Surprisingly, even the small-sized models (e.g. TinyLlama-1.1B) can achieve a score of 0.89 in the scenario of copying in unstructured context. While the scores of larger models are no more than this value and the best model in our evaluation is only 0.9. For the scenario of copying in structured context, the results are also unrelated to the models' sizes. The ChatGLM3-6B model (base-version) gets the best score of 0.70, which is higher than the models having 5x-10x size (e.g. LLaMA-65B).

(3) For the learning ability, larger models usually have better performance than smaller ones. The designed rules in format problems are easy to learn, and there are no huge gaps between small-sized models and large-sized models. The middle-sized model (Baichuan2-13B) gets the best score of 0.75 in this scenario. However, for more challenging rules, the gaps between small-sized models and large-sized models become more obvious. The highest scores of small-sized and large-sized models are 0.5 v.s. 0.75, 0.25 v.s. 0.45, and 0.4 v.s. 0.5 in order, statistics, and complex rules' scenarios respectively. Especially, the largest model in our evaluation (LLaMA-65B) gets two of them.



Figure 3: The scores in the pretraining stage of TinyLlama-1.1B with 3T tokens.



Figure 4: The scores in the pretraining stage of Baichuan2-7B with 2.6T tokens.



Figure 5: Baichuan2-7B's ICL ability and knowledge. The ICL ability of the model is acquired in the early stage of pretraining, while the knowledge is acquired in the whole pretraining stage.

3.3 How does Pretraining Token Number Effect ICL Ability?

We also examine how the ICL ability evolves during the pretraining stage with the released pretraining checkpoint of TinyLlama-1.1B, Baichuan-7B, and Amber-7B. Due to space limit, the results of Amber-7B models' checkpoints are provided in Appendix C.3. The evaluation results of TinyLlama-1.1B and Baichuan2-7B are shown in Figure 3 and Figure 4 respectively. Combining the two figures, we observed that:

(1) All the in-context learning abilities have a rapid growth stage before 200B tokens, and then the abilities grow slowly and even stop growth. We guess that the models are mainly to learn more about grammar, world knowledge, and complex skills after acquiring ICL abilities. To evidence our thoughts, we test the TriviaQA (Joshi et al., 2017) and MMLU (Hendrycks et al., 2020) datasets in the same checkpoints, as shown in Figure 5. We find that after training 220 tokens, the ICL abilities nearly no increase, while knowledge of models continue to increase.

(2) The copying ability emerges in the very early stages of pretraining. The ability to copy information from **unstructured context** arrives at the highest before 10B tokens and remains stable until the end of pretraining. Besides, the ability to copy information from **structured context** arrives at the highest score of about 0.53 at the 419B tokens. It is unstable and fluctuates multiple times during the subsequent training stages. We suppose that it may be due to the scarcity of structured data in the pretraining data, making it challenging to retain the ability to copy target information in these scenarios.

(3) The learning ability to format rules and order rules shows little improvement after 200B tokens. Regarding the **format rule learning**, models need to learn rules to judge formats, follow custom formats, and convert one format to another. These rules are fundamental and easy to learn. However,

we find the models' predictions will be interfered 310 with by their inherent preferences (Details will be 311 discussed in Section 3.4.). We speculate that during 312 the main pertaining stage models will continuously 313 acquire new knowledge and adjust their inherent preferences. This makes the scores unstable and 315 prevents models from getting better results. For 316 the order rule learning, the scores are stable but 317 nearly no improvement after the early pretraining stage (e.g.180B tokens for TinyLlama-1.1B). We 319 believe it is due to the lack of this aspect in the pretraining data. Models struggle to learn a new 321 rule that they have never encountered before. 322

323

324

325

328 329

331

338

339

340

342

343

345

346

(4) The learning ability to learn statistics rules and complex rules is continually increasing in the whole pretraining stage, indicating that continued pretraining is useful for these tasks. In particular, there is a significant increase in statistics for Baichuan-7B but only a slight improvement in TinyLlama-1.1B. We guess that maybe models only learn to predict the next token based on a small number of previous tokens in the early stages of pretraining. While, during the later stage of pretraining, models learn to use more tokens to predict the next token more accurately. Small-sized models may not have enough attention points capacity for later pretraining, which will be discussed in detail in Section 3.4.

3.4 Detailed Discussion for Other Factors Effect ICL Ability.

We find some interesting phenomenons in our evaluation results and regard these phenomena to four aspects: distinguishing ability, inherent preferences, attention points capacity, and tokenizer. We also analyze these phenomena with some bad cases. Due to the limitation of pages, we put the bad cases on the Appendix C.1.

347Distinguishing AbilitySimilar strings can cause348interference for humans as well as for LLMs. As349depicted in Figure 6, in the dictionary search task,350the accuracy of the same model is adversely af-351fected when there are numerous similar keys in the352dictionary. More similar keys make the scores drop353more, while models with the stronger distinguish-354ing ability (e.g. Baichuan2-7B) drop less. The355results suggest that the excessive presence of sim-356ilar strings makes the model chaotic, causing it357to struggle to distinguish between different keys.358This phenomenon indicates that models with weak359distinguishing ability may make mistakes when



Figure 6: Performance changes when there are more similar strings in the in-context examples.



Figure 7: The proportion of three answer types in the pretraining stage of Baichuan2-7B. Green means copying right value, blue means copying incorrect values, and pink means genrating random values. We find the model's choices are continuously changing in the pre-training stage.

extracting information in complex in-context environments. 360

361

362

363

364

366

367

368

369

370

371

372

373

374

375

376

377

379

380

381

We looked into the reasons for errors in certain cases and found two types. First, the model sometimes selects the incorrect value from the dictionary. Second, it occasionally generates a random string that is not present in the dictionary. We further analyze the proportion of right results and the two error types in the pretraining stage of Baichuan2-7B, as shown in Figure 7. We find the model's choices of the two types are continuously changing. That indicates the distinguishing ability is unstable in the pretraining stage.

Inherent Preferences We observe that some models exhibit unusually weak performance in tasks such as format check or format cloning. Figure 8 shows the performance of the format check task. This task is a classification task with six different labels: "jsonl", "csv", "tuple", "yaml", "table", and "xml". From the figure, we can find that ChatGLM3-6B-Chat can get a score approaching 0.7, while lots of models are lower than the random scores. To our surprise, LLaMA-65B gets 0.0 in this task.



Figure 8: Performance of format check task.

We further give a deeper analysis of the bad cases, and we find that most models tend to respond with "json" as their prediction, but we even don't have this label. We suppose that such heavy inherent preferences of these models may come from their pretraining data distribution, making the model cannot adhere to the pre-defined formats or rules presented in the in-context examples.

Furthermore, we find the chat version of some models such as ChatGLM3-6B, InternLM-7B, and Mistral-7B have obvious improvements to their base version in this task. We think this might be due to the instruction learning process can reduce the impact of the models' inherent preferences to some extent.

Attention Points Capacity In the count & navigation task, the "1-dim" setting requires models to analyze and count only two elements ("up" and "down"), whereas the "2-dim" setting involves analyzing and counting four elements ("up", "down", "right" and "left"). We have observed that the scores for the "1-dim" setting are significantly higher than those for the "2-dim" setting, as illustrated in Table 3.

We believe this phenomenon occurs because models face difficulties in effectively utilizing a larger number of tokens within the context to predict the next token. In other words, for models 411 to accurately predict results, they need to pay at-412 tention to multiple points within the context. And 413 we call this Attention Points Capacity. However, 414 many models tend to rely on only a few tokens for 415 predicting the next token, thereby failing to incor-416 porate all the available information simultaneously. 417 To validate our hypothesis, we conducted a further 418 analysis of the relationship between the number 419

Table 3: The scores of count & navigation task. We split the tasks by task types.

M. 1.1	Co	unt	Navigation		
NIOdel	1-dim	2-dim	1-dim	2-dim	
TinyLlama-1.1B	0.17	0.03	0.27	0.03	
Yi-6B	0.40	0.00	0.43	0.03	
LLaMA2-7B	0.47	0.07	0.27	0.13	
Mistral-7B-v0.1	0.77	0.43	0.47	0.10	
Qwen-7B	0.60	0.10	0.53	0.10	
Baichuan2-7B	0.77	0.07	0.33	0.07	



Figure 9: The scores of count & navigation task. We split the task by the statistics elements' number. More elements indicate more need for attention points.

of elements and accuracy, as depicted in Figure 9. The figure demonstrates a significant decrease in accuracy as the number of elements increases.

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

Tokenizer In our evaluation, we observe that the tokenizer has a significant impact on our results and can affect our adjustment of task difficulty. From a human perspective, we can perceive text at the character level, word level, or sentence level, allowing us to easily distinguish individual numbers, letters, words, and sentences. However, language models process inputs at the token level, which presents a different viewpoint compared to ours.

In Figure 10, we illustrate three types of bad cases caused by tokenization that we find can lead to confusion for LLMs. "Token fracture" occurs when a continuous sequence is split but the encoded tokens of the segmented fragments do not match those of the original sequence; "Token insert" refers to the fact that the special symbols that we can easily overlook also may require 1-3 tokens for encoding; "Token replace" indicates that several consecutive characters can be encoded as a single token, resulting in different encoded tokens for the reversed string compared to the original one.



Figure 10: The three tokenization phenomena in which token views are different from human views.

4 Related Work

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

In-Context Learning The generalization ability of language models has always been a goal pursued by researchers, and the discovery of in-context learning ability is a milestone in achieving this goal. It was discovered in GPT-3 (Brown et al., 2020) that the model can implement custom outputs based on instructions and examples. In-context learning can enable models to adapt to new tasks and even learn new knowledge and rules without updating model parameters. The phenomenon of in-context learning has attracted widespread attention from researchers to explore (Lu et al., 2021; Nie et al., 2022; Ye et al., 2022; Min et al., 2022; Liu et al., 2023a), explain (Xie et al., 2021; Akyürek et al., 2022; Han et al., 2023; Li et al., 2023), enhance (Min et al., 2021; Chen et al., 2021b; Yang et al., 2023), and apply (Dua et al., 2022; Shridhar et al., 2022; Wu et al., 2023) it. Chain-of-thought (CoT) reasoning is special in-context learning, divided into two modes: few-shot CoT (Wei et al., 2022) and zero-shot CoT (Kojima et al., 2022). Few-shot prompts is an important part of CoT, and many studies (Wang et al., 2022; Zhou et al., 2022; Fu et al., 2022; Lyu et al., 2023; Shum et al., 2023; Diao et al., 2023) focus on selecting better examples to promote the application of CoT. we propose an ICLEval benchmark for measuring in-context learning abilities. To the best of our knowledge, it's the first work to estimate ICL abilities comprehensively.

Model Evaluation After the emergence of the 475 large model, to comprehensively understand the 476 boundaries and behaviors of its abilities, re-477 searchers tested the model from multiple aspects 478 such as language ability, knowledge, professional 479 ability, theory of mind, and security. The test-480 ing of model language abilities can utilize vari-481 ous traditional NLP task datasets, including lan-482 guage comprehension (Qin et al., 2023a), lan-483 guage generation (Qin et al., 2023a), and mul-484 tilingualism (Lai et al., 2023). The testing of 485 model knowledge includes common sense knowl-486 edge (Clark et al., 2018; Bisk et al., 2020; Zellers 487 et al., 2019; Mihaylov et al., 2018), factual knowl-488 edge (Kwiatkowski et al., 2019; Lin et al., 2021; 489 Berant et al., 2013), technical knowledge (Yuan 490 et al., 2023), etc. The Achievement test for model 491 specialty includes mathematics (Cobbe et al., 2021; 492 Ling et al., 2017; Hendrycks et al., 2021), coding 493 (Austin et al., 2021; Chen et al., 2021a), medicine 494 (Thirunavukarasu et al., 2023), etc.In (Bubeck 495 et al., 2023), theory of mind is considered to test 496 GPT-4(OpenAI, 2023). The test of model secu-497 rity includes stability, ethics, biases, and hallu-498 cination. Among them, many benchmarks can 499 perform relatively comprehensive evaluation on 500 the models, such as HELM(Liang et al., 2022), 501 MMLU (Hendrycks et al., 2020), C EVAL(Huang 502 et al., 2023), AGIEval(Zhong et al., 2023), BIG-503 Bench(Srivastava et al., 2022), etc. However, most 504 of the evaluations here need to involve a variety of 505 different basic capabilities, and the quality of the 506 test results is affected by a variety of factors. Our 507 ICLEval benchmark decouples the evaluation of 508 the in-context learning ability from other abilities. 509

5 Conclusion

In this work, we introduce an ICLEval benchmark for measuring the in-context learning ability of large language models. We classify it into copying ability and learning ability, and design 12 evaluation tasks. We find that model size is an important but not the sole determinant of ICL ability. Also, we observe that ICL abilities, particularly copying ability have a quick increase in the very early pretraining stage. Furthermore, we discover that the ICL abilities are influenced by distinguishing ability, inherent preferences, attention points capacity, and tokenizer. That indicates that we need to consider these factors if we would like to train a model with strong in-context ability. 510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

526

- 531 532 533 534 535 536 537 538
- 539 540
- 541
- 542 543 544
- 545 546 547
- 548 549
- 552
- 553 554
- 555 557
- 558 559 560

561

563

- 564
- 565 566

567 568

570

571

573

574

When designing the ICLEval evaluation, we minimize the need for other abilities of the model as much as possible. However, we still can't guarantee that the testing tasks will not be affected by the language abilities and internal knowledge of the model itself.

References

Limitations

- Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. arXiv preprint arXiv:2211.15661.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. arXiv preprint arXiv:2108.07732.
- Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In Proceedings of the 2013 conference on empirical methods in natural language processing, pages 1533-1544.
- Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pages 7432-7439.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. Advances in neural information processing systems, 33:1877-1901.
- Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. arXiv preprint arXiv:2303.12712.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021a. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374.
- Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2021b. Meta-learning via language model in-context tuning. arXiv preprint arXiv:2110.07814.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind

Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv preprint arXiv:1803.05457.

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168.
- Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-ofthought for large language models. arXiv preprint arXiv:2302.12246.
- Dheeru Dua, Shivanshu Gupta, Sameer Singh, and Matt Gardner. 2022. Successive prompting for decomposing complex questions. arXiv preprint arXiv:2212.04092.
- Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. arXiv preprint arXiv:2210.00720.
- Chi Han, Ziqi Wang, Han Zhao, and Heng Ji. 2023. In-context learning of large language models explained as kernel regression. arXiv preprint arXiv:2305.12766.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874.
- Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiavi Lei, et al. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. arXiv preprint arXiv:2305.08322.
- Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada. Association for Computational Linguistics.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. Advances in neural information processing systems, 35:22199-22213.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark

632

for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.
- Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning. *arXiv* preprint arXiv:2304.05613.
 - Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. 2023. Transformers as algorithms: Generalization and stability in in-context learning.
 - Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.
 - Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.
 - Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.
- Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, Richard Fan, Yi Gu, Victor Miller, Yonghao Zhuang, Guowei He, Haonan Li, Fajri Koto, Liping Tang, Nikhil Ranjan, Zhiqiang Shen, Xuguang Ren, Roberto Iriondo, Cun Mu, Zhiting Hu, Mark Schulze, Preslav Nakov, Tim Baldwin, and Eric P. Xing. 2023b. Llm360: Towards fully transparent open-source llms.
- Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.
- Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-ofthought reasoning. *arXiv preprint arXiv:2301.13379*.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.
- Sewon Min, Mike Lewis, Luke Zettlemoyer, and Han-687 naneh Hajishirzi. 2021. Metaicl: Learning to learn in 688 context. arXiv preprint arXiv:2110.15943. Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettle-691 moyer. 2022. Rethinking the role of demonstra-692 tions: What makes in-context learning work? arXiv 693 preprint arXiv:2202.12837. 694 Feng Nie, Meixi Chen, Zhirui Zhang, and Xu Cheng. 695 2022. Improving few-shot performance of language 696 models via nearest neighbor calibration. arXiv 697 preprint arXiv:2212.02216. 698 OpenAI. 2023. Gpt-4 technical report. 699 Tianduo Wang Peiyuan Zhang, Guangtao Zeng and Wei 700 Lu. 2023. Tinyllama. 701 Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao 702 Chen, Michihiro Yasunaga, and Diyi Yang. 2023a. Is 703 chatgpt a general-purpose natural language process-704 ing task solver? arXiv preprint arXiv:2302.06476. 705 Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, 706 Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, 707 Chaojun Xiao, Chi Han, et al. 2023b. Tool 708 learning with foundation models. arXiv preprint 709 arXiv:2304.08354. 710 Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. 711 Know what you don't know: Unanswerable questions 712 for squad. arXiv preprint arXiv:1806.03822. 713 Siva Reddy, Dangi Chen, and Christopher D Manning. 714 2019. Coga: A conversational question answering 715 challenge. Transactions of the Association for Com-716 putational Linguistics, 7:249–266. 717 Kumar Shridhar, Alessandro Stolfo, and Mrinmaya 718 Sachan. 2022. Distilling multi-step reasoning ca-719 pabilities of large language models into smaller mod-720 els via semantic decompositions. arXiv preprint 721 arXiv:2212.00193. KaShun Shum, Shizhe Diao, and Tong Zhang. 2023. 723 Automatic prompt augmentation and selection with chain-of-thought from labeled data. arXiv preprint 725 arXiv:2302.12822. Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the 730 imitation game: Quantifying and extrapolating the capabilities of language models. arXiv preprint 732 arXiv:2206.04615. 733 InternLM Team. 2023. InternIm: A multilingual 734 language model with progressively enhanced capabilities. https://github.com/InternLM/ InternLM. 737

841

793

794

Arun James Thirunavukarasu, Refaat Hassan, Shathar Mahmood, Rohan Sanghera, Kara Barzangi, Mohanned El Mukashfi, and Sachin Shah. 2023. Trialling a large language model (chatgpt) in general practice with the applied knowledge test: observational study demonstrating opportunities and limitations in primary care. *JMIR Medical Education*, 9(1):e46599.

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

756

758

762

763

764

765

766

767

768

770

773

775

776

777

781

782

784

785

786

787

790

791

- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.
- Zhenyu Wu, YaoXiang Wang, Jiacheng Ye, Jiangtao Feng, Jingjing Xu, Yu Qiao, and Zhiyong Wu. 2023. Openicl: An open-source framework for in-context learning. *arXiv preprint arXiv:2303.02913*.
- Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.
- Jiaxi Yang, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Iterative forward tuning boosts in-context learning in language models. *arXiv preprint arXiv:2305.13016*.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.
- Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Ves Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2022.
 Complementary explanations for effective in-context learning. *arXiv preprint arXiv:2211.13892*.
- Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, and Songfang Huang. 2023. How well do large language models perform in arithmetic tasks? *arXiv preprint arXiv:2304.02015*.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.
- 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 arXiv:2304.06364.
- Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2022.

Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A ICLEval Benchmark

A.1 Data Source

The data for our ICLEval Benchmark is sourced from various places, including Wikipedia, a common noun vocabulary, GSM8K, AQuA, BIGBench, and generation context from ChatGPT.

For paragraph-level and sentence-level data, we collected information from Wikipedia. Initially, we randomly selected 100 articles from Wikipedia to form the basis of natural language paragraphs. These paragraphs serve as the core content for the string completion task, which tests the models' ability to accurately copy information in unstructured context. It's important to note that we processed these paragraphs to ensure that the models hadn't encountered them during the pre-training stage. Additionally, we extracted all the sentences containing 5-30 words from these Wikipedia paragraphs to create sentence-level data. This data is used for tasks such as dictionary search, order adjustment, and de-duplication.

For word-level data, we gathered 1526 common nouns from a common noun vocabulary. Furthermore, we included all numbers, uppercase letters, and lowercase letters as character-level data. These datasets primarily focus on tasks related to solving order-related problems.

In addition, we sampled 50 examples from the GMS8K and AuQA datasets, respectively, to create the format cloning task. It's important to note that we are not concerned with the accuracy of the math questions themselves, but rather with the format of the predictions. Furthermore, we generated virtual data in the "person" and "company" domains using ChatGPT, which serves as the foundational content for tasks such as format check and format conversion.

For tasks related to statistics problems, we drew inspiration from the "navigate" and "long_context_integration" tasks in BIG-Bench(Srivastava et al., 2022). Additionally, we utilized the data from the "list_numbers" task to create our numbers' rules task.

A.2 Tasks

We partition our tasks into two categories: testing copying abilities and testing learning abilities. The

890

first two tasks focus on assessing copying abilities, while the remaining tasks evaluate learning abilities.

String Completion task is specifically designed to test copying abilities in natural language context scenarios. In this task, models are required to predict the second half of a string given the first half. For each testing sample, we randomly select a lengthy paragraph from Wikipedia as the basis for a natural language context. To ensure that the model has not encountered the main entity mentioned in the paragraph before, we replace it with a 16-character hash string. Furthermore, we split the last hash string in the paragraph into two 8character halves, and the target for the model is to predict the subsequent 8 characters. We have designed this task because it closely resembles the pre-training task and serves as a fundamental form of assessing copying ability. An example of this task can be seen in Figure 17.

Dictionary Search task involves predicting a value based on a given key from a set of key-value pairs. We explore two different forms of this task. The first form consists of short keys and long values, while the second form features short values but longer, more intricate keys. The samples in this task can be seen in Figure 17.

In the first form, we provide 20 key-value pairs with varying lengths as in-context examples. The key is a 6-character random hash string, while the value is a random sentence sampled from Wikipedia.

The second form might be more difficult. We provide 10 key-value pairs that involve number calculations as in-context examples. The key is a long number calculation string with more than 20 numbers and operators. Moreover, to examine the impact of similar keys, we establish three levels for this task: "all-similar," "half-similar," and "nonsimilar." These levels determine the number of keys that share a similarity with the prediction target.

Format Check is a classification task to distinguish which class the current format is. We chose six common formats: JSONL, CSV, Triple-Tuple, YAML, Markdown-Table, and XML. Then we will randomly generate 6-shot examples using the six formats respectively as the in-context for each testing sample. Each sample is a random format and belongs to the six formats. The samples in this task can be seen in Figure 18.

Format Cloning is a generation task to follow

the customized formats. We have designed five customized formats for the CoT output of GSM8K (Cobbe et al., 2021) and the multi-choice output of AQuA (Ling et al., 2017). And we randomly sample 5-shot examples from this dataset. In these tasks, we check the correctness of the output format without focusing on the correctness of the results themselves. The samples in this task can be seen in Figure 18.

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

Format Conversion is a generation task to convert a source format to a target format and keep the content consistent. We also use the six formats mentioned in the "format-check" task. To explore more complex scenarios, we set four forms in this task: "single", "multi", "transfer" and "mix". We have two domains ("people" and "company") of data as content. "single" form means only converting one item, and the domain is consistent between in-context examples and testing samples. "multi" form will have 1-5 items based on the "single" form. While the "transfer" form also converts one item the in-context examples and testing samples are from different domains. The "mix" form might most difficult which means there will be multiple items and both two domains' data as content. The samples in this task can be seen in Figure 19.

Order Check is a classification task to judge if the order of elements in one string is reversed. We set 8-shot examples and each example has two strings. If the two strings have the same order, the label is False, while if the two strings have reversed order, the label is True. We will randomly generate different 8-shot examples for every testing sample, and the proportion of positive and negative examples is also random. We have word-level and character-level settings for different element granularity. The sample in this task can be seen in Figure 20.

Order Adjustment is a generation task to generate sequence, reverse, or specify form string given origin string. For the sequence setting, the model is only required to copy the input text without any changes. For the reverse sequence setting, the model needs to output the reversed version of the original text. For the specific sequence setting, the model must learn to identify and execute a specified index operation on the input text. We randomly generate 5-shot examples for each testing sample, and hope models can learn and apply the order adjustment rules to the testing sample. To enhance diversity, we experiment with characterlevel, word-level, and sentence-level input-output pairs for different element granularity. The sample in this task can be seen in Figure 20.

945

946

947

949

952

953

955

957

960

961

962

964

965 966

967

970 971

972

973

974

975

976

977

979

983

987

991

992

Count & Navigation is a generation task to return a dictionary depending on the in-context information. The model is given a list of dictionary nouns (up, down, right, and left) as input. The count task requires the model to count the number of occurrences of all the nouns, while the navigation task involves considering the final point relative to the initial point. We set "easy" mode and "hard" mode for this task. The "easy" mode only has two dictionary nouns while the "hard" mode has four. The sample in this task can be seen in Figure 21.

Relation Analysis task provides the model with a relation graph as input and requires it to output all the nodes connected to a chosen node. Each of these tasks includes 5-shot examples as in-context information. We have adjusted the nodes or sides in the relation graph for this task, to introduce variations for different models. The sample in this task can be seen in Figure 21.

Duplication Check a classification task to judge if there are repeated elements in the string. If there are repeated elements in the string, the label is True, else the label is False. We use the same setting as the "order-check" task, which has 8-shot examples for every testing sample and has word-level and character-level element granularity. The sample in this task can be seen in Figure 21.

De-Duplication is a generation task to find or remove repeat elements from a string. Finding repeat elements means only outputting the repeat element in one string while removing repeat elements means outputting a whole string without any repeat elements. We use sentence-level settings for finding repeat elements form and use characterlevel and word-level settings for removing repeat elements form. As previous setting, we randomly generate 5-shot examples for each testing sample. The sample in this task can be seen in Figure 21.

Numbers' Rules is a task that consists of multiple groups of number pairs. Each pair comprises an input list and an output list, with an internal rule governing the transformation from input to output. With multi-group examples as the in-context information, the model needs to learn the underlying rule and predict the output list for a new input list. We have collected task data for this evaluation from the "list_functions" task in BIGBench. In this task, there are 250 different rules, and some rules are even very hard for humans to find. The samples in this task can be seen in Figure 16.

Table 4: The max generation length and split string for different tasks. We set the max generation length for testing effectiveness.

Task Name	Split String	Max Len.
	Spirtstring	
Copying	all nunctuation marks	10
dictionary search	an punctuation marks	10
- short key	\n	75
- long and similar key	\n	12
Learning		_
format check	\n	5
format cloning	\nQuestion:	196
format conversion	\nInput:	256
order check	\n	5
order adjustment		
- character-level	\n	50
- word-level	\n	50
- sentence-level	\nInput:	256
count & navigation	\n	30
relation analysis	\nInput:	60
duplication check	\n	5
de-duplication		
- character-level	\n	30
- word-level	\n	30
- sentence-level	\n	60
numbers' rules	\n	50

A.3 Metrics

Considering the effectiveness of ICLEval, we set different max generation length limitations for various tasks. Meanwhile, we set different split strings for processing models' responses to final predictions. Both the settings are shown in Table 4. Setting the two parameters is because the base-version model will not stop when giving in-context examples. The max generation length will make tested models stop their generation in the suit position. And we will regard the content in front of the split strings as the right predictions. Then we will strip the blank space at both ends of the right predictions.

B Evaluated Models

We show the details of our evaluated models in this section, and the models are listed in Table 5.

GPT-Neo-1.3B is a transformer model with 1.3 billion parameters, trained on the Pile, a curated dataset created by EleutherAI specifically for training this model.

Phi-1.5 is a transformer model with 1.3 billion parameters, trained by Microsoft Research. It is

998

999

1000

1001

1002

1003

1004

1006

1008

1009

1010

1012

1013

1014

1015

1016

1017

1018

1019

Model	Version	Parameters	Training Data
GPT-Neo	Base	1.3B	0.38T Tokens
GPT-J	Base	6B	0.4T Tokens
TinyLlama	Base	1.1B	3T Tokens
		7B	1T Tokens
LL oMA1	Base	13B	1T Tokens
LLaWAI	Dase	34B	1.4T Tokens
		65B	1.4T Tokens
II aMAO	Dece/Chat	7B	2T Tokens
LLaMA2	Base/Chat	13B	2T Tokens
Mistral	Base/Chat	7B	_
ChatGLM3	Base/Chat	6B	-
VI	Deco/Chat	6B	3T Tokens
11	Dase/Cliat	34B	3T Tokens
Baichuan?	Base/Chat	7B	2.6T Tokens
Daicifuali2	Dase/Cliat	13B	2.6T Tokens
Owen	Base/Chat	7B	>2.4T Tokens
Qwen	Dase/Cliat	14B	>3T Tokens
Phi-1.5	Base	1.3B	_
InternLM	Base/Chat	7B	>1T Tokens
Skywork	Base	13B	3.2T Tokens
Amber	Base	7B	1.2T Tokens

Table 5: Models evaluated in our work

specialized in basic Python coding and is augmented with various NLP synthetic texts.

1020

1021

1022

1023

1024

1025

1026

1027

1028

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1042

1043

1044

1045

1046

1047

1048

1049

TinyLlama project (Peiyuan Zhang and Lu, 2023) aims to pretrain a 1.1 billion parameter Llama model on 3 trillion tokens. The model's intermediate checkpoints, corresponding to about 10 billion tokens, are open-sourced every 5,000 steps. This project is released by the StatNLP Research Group of Singapore University of Technology and Design.

GPT-J is a transformer model trained with 6 billion parameters. It is trained by EleutherAI.

ChatGLM3-6B is the latest open-source model in the ChatGLM series, utilizing the General Language Model (GLM) architecture with 6 billion parameters. It was released by the Knowledge Engineering Group (KEG) & Data Mining at Tsinghua University.

YI-6B and YI-34B belong to the Yi series models, trained from scratch by 01.AI. These models are based on a 3 trillion multilingual corpus and have 6 billion and 34 billion parameters, respectively.

LLaMA1-7B and LLaMA1-13B belong to the Llama series models released by Meta AI. They are trained on 1 trillion tokens.

LLaMA1-30B and LLaMA1-65B belong to the Llama series models released by Meta AI. They are trained on 1.4 trillion tokens.

LLaMA2-7B and LLaMA2-13B belong to the

Llama 2 series models, which are a collection of pre-trained and fine-tuned generative text models ranging in scale from 7 billion to 70 billion parameters. These models are auto-regressive language models trained on 2 trillion tokens and were released by Meta AI. 1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1071

1072

1073

1074

1075

1076

1078

1080

1081

1082

1084

1085

1087

1088

1089

1090

Mistral-7B-v0.1 is a pre-trained generative text model with 7 billion parameters. It was released by Mistral AI.

Skywork-13B was trained on a high-quality cleaned dataset consisting of 3.2 trillion multilingual data, mainly in Chinese and English, including code. It was released by Skywork AI.

Baichuan2-7B and Baichuan2-13B belong to the Baichuan2 series models, which are opensource and commercially usable large-scale language models developed by Baichuan Intelligence. They are trained on a high-quality corpus with 2.6 trillion tokens. Baichuan Intelligence also opensources intermediate checkpoints of Baichuan2-7B every 220 billion tokens.

Qwen-7B and Qwen-14B are versions of the large language model series called Qwen (Tongyi Qianwen) proposed by Alibaba Cloud. These Transformer-based large language models have 7 billion, 14 billion, and 72 billion parameters, respectively. They are trained on a large volume of data, including web texts, books, code, etc.

InternLM-7B (Team, 2023) is an open-sourced base model with 7 billion parameters, tailored for practical scenarios.

Amber-7B is a 7 billion parameter language model with the same architecture as LLaMA-7B, trained on Arxiv, Book, C4, Refined-Web, Star-Coder, StackExchange, and Wikipedia. It was released by LLM360 (Liu et al., 2023b), which is an initiative for comprehensive and fully open-source language models. And it provides open-source 360 intermediate checkpoints of Amber-7B.

C Experiments Results

C.1 Bad Cases Analysis

We list some bad cases in this sub-section.

Many models can arrive at about 0.89 scores in 1092 the entity completion task, but it's hard for them to 1093 get higher scores. We analyzed the bad case and 1094 found that there are some entities (hash string) that 1095 have token-fracture phenomenons when splitting 1096 them. As shown in Figure 11, the tokens of string 1097 "521351caba13f591" are not equal to the tokens 1098 of "521351ca" add the tokens of "ba13f591". This 1099

The Baptist Mission Union censured 521351caba13f591 for abandoning his post without authorization and going to Rangoon; Calista carried on the mission work, assisted by her son Brainerd and daughter Calista and their spouses, who returned from America after their studies.\nIn 1872, long after the death of 521351ca			
"521351caba13f591" to tokens: [29871, 29945, 29906, 29896, 29941, 29945, 29896, 29883 , 5363 , 29896, 29941, 29888, 29945, 29929, 29896] "521351ca" to tokens: [29871, 29945, 29906, 29896, 29941, 29945, 29896, 1113]			
LLaMA2-7B:	13f591		
Mistral-7B:	13f591		
Yi-6B:	ab13f591		
Baichuan2-7B:	13f591		
Qwen-7B:	13f591		
Yi-34B:	13f591		
LLaMA-65B: 13f591			
LLaMA-65B:	151571		

Figure 11: A bad case of entity completion task and order adjustment task. These two bad cases are selected from the predictions of LLaMA2-7B.

makes it difficult for models to finish this task.

We give a bad case of dictionary search task in Figure 12. Similar examples confuse the models, and only the LLaMA-65B model finds the right value.

We give a bad case of format check task in Figure 13. Lots of models predict the results depending on the models' inherent preferences rather than the in-context examples. The rules in this task are easy to learn so that small-sized models (e.g. TinyLlama) can predict the right labels. However, the larger models (e.g. LLaMA-65B) might predict wrong labels due to their inherent preferences.

We give two bad cases of count & navigation task in Figure 15. Models have good performance when there are several elements to statistics. The performance has a huge drop when the number of elements becomes greater.

We give two bad cases of numbers' rules task in Figure 16. Some rules are even very difficult for humans. We chose two hard samples and showed the mean human score in this figure.

C.2 Detailed Results

We list the detailed results in Table ??. Each col-
umn is the results of all the models we tested for
one task. We list the mean results of all the tasks in
the last column of this table. We also put the maxi-
mum and minimum results in the last two rows of
the table.1123
1124
11251123
1124
11251124
1125

1122

1129

C.3 Results in Pretraining Stage

In this section, we tested the Amber-7B models1130with the multi checkpoints, trained on 1.2T tokens,
and saved about every 3.5B tokens. The ICL abili-
ties scores are shown in Figure 14. And we can get1131similar conclusions as the results of Baichuan2-7B1134and TinyLlama-1.1B.1135

ng (Dictionary Search)			
148 * 348 D 159 * 134 D 452 ⊄ 48 * 343 D 159 * 134 D 452 ⊄ 148 * 348 D 159 * 134 D 452 ⊄ 145 * 348 D 159 * 134 D 452 ⊄ 148 * 348 D 159 * 134 D 452 ⊄ 148 * 348 D 159 * 134 D 452 ⊄	 ≥ 94 * 891 D 88 D 526 * 97 D 607 2 -35939 94 * 891 D 88 D 526 * 97 D 607 2 -35894 ≥ 94 * 391 D 88 D 526 * 97 D 607 2 -36439 ≥ 94 * 891 D 88 D 526 * 97 D 607 2 -36662 ≥ 94 * 891 D 88 D 526 * 97 D 607 2 -35936 ≥ 94 * 891 D 88 D 526 * 97 D 607 2 -35936 ≥ 94 * 891 D 88 D 526 * 97 D 607 2 -36002 		
$73 \ 55 \ *98 \ 858 \ *894 \ *741 \ *51 \ *86 \ column{$<}{3} \ 72 \ 94 \ 758 \ 23 \ *148 \ *348 \ 159 \ *134 \ 94 \ *891 \ 152 \ column{$<}{3} \ 94 \ *891 \ 98 \ 526 \ *97 \ 9607 \ 2-7739 \ 72 \ 55 \ *98 \ 858 \ *894 \ *741 \ *51 \ *86 \ column{$<}{3} \ 72 \ 94 \ 758 \ 23 \ *148 \ *348 \ 159 \ *134 \ 9452 \ column{$<}{3} \ 94 \ *891 \ 88 \ 526 \ *97 \ 9607 \ 2-7739 \ 72 \ 55 \ *98 \ 858 \ *894 \ *741 \ *51 \ *86 \ column{$<}{3} \ 72 \ 94 \ 758 \ 23 \ *148 \ *348 \ 159 \ *134 \ 9452 \ column{$<}{3} \ 94 \ *891 \ 88 \ 526 \ *97 \ 9607 \ 2-739 \ 72 \ 355 \ *98 \ 858 \ *894 \ *741 \ *51 \ *86 \ column{$<}{3} \ 72 \ 94 \ 758 \ 23 \ *148 \ *348 \ 159 \ *134 \ 9452 \ column{$<}{3} \ 94 \ *891 \ 88 \ 526 \ *97 \ 9607 \ 2-35940 \ 72 \ -35940 \ 73 \ 55 \ *89 \ 88 \ 894 \ *741 \ *51 \ *86 \ column{$<}{3} \ 72 \ 94 \ 758 \ 23 \ *148 \ *348 \ 159 \ *134 \ 9452 \ column{$<}{3} \ 94 \ *891 \ 88 \ 526 \ *97 \ 9607 \ 2-35940 \ 72 \ -35939 \ 73 \ 55 \ *98 \ 88 \ 884 \ *741 \ *51 \ *86 \ column{$<}{3} \ column{$<}{3} \ 83 \ 83 \ 83 \ 159 \ *134 \ 9452 \ column{$<}{3} \ 94 \ *891 \ 88 \ 526 \ *97 \ 9607 \ 2-35939 \ 73 \ 55 \ *98 \ 894 \ *741 \ *51 \ *86 \ column{$<}{3} \ column{$<}{3} \ *148 \ *348 \ 159 \ *134 \ 9452 \ column{$<}{3} \ 94 \ *891 \ 88 \ 526 \ *97 \ 9607 \ 2-35939 \ 73 \ 55 \ *98 \ 88 \ 894 \ *741 \ *51 \ *86 \ column{$<}{3} \ 72 \ 94 \ 758 \ 23 \ *148 \ *348 \ 159 \ *134 \ 9452 \ column{$<}{3} \ 94 \ *891 \ 88 \ 526 \ *97 \ 9607 \ 2-35939 \ 73 \ 55 \ *98 \ 88 \ 884 \ *741 \ *51 \ *86 \ column{$<}{3} \ 72 \ 94 \ 878 \ 23 \ *148 \ *348 \ 83 \ 159 \ *134 \ 9452 \ column{$<}{3} \ 94 \ *891 \ 88 \ 88 \ 88 \ 88 \ 88 \ 88 \ 88 \$			
LLaMA-65B	: -35939		
Label:	-35939		
Metrics: we will check label number.	if the prediction number is exact match with the		
	ag (Dictionary Search) 148 * 348 0 159 * 134 0 452 % 48 * 343 0 159 * 134 0 452 % 148 * 348 0 159 * 134 0 452 % 148 * 348 0 159 * 134 0 452 % 148 * 348 0 159 * 137 0 452 % 148 * 348 0 159 * 134 0 452 %		



	Learning (Form	at Check)	
Prompt: Input: [Index name age city 1 Ava Hill 31 Portland Output: table Input: Index,name,age,city 1,David Wilson,29,Boston Output: csy		Output: yaml Input: {"name": "Grad Output: jsonl Input: Index name ag 1 Brooklyn Wi Output:	ce Morgan", "age": 30, "city": "St. Louis"} e city ilson 35 Tampa
Input: (Landon Smith, age, 31) (Landon Smith, city, New Orleans) Output: tuple Input: <person> <name>Alexander Harris</name> <age>33</age> <city>Philadelphia</city> </person> Output: xml Input: person: name: Charlotte Adams age: 35		Tiny Llama: LLaMA2-7B: Mistral-7B: Yi-6B: Baichuan2-7B: Qwen-7B: Yi-34B: LLaMA-65B: Label:	tablecsvtablecsvcsvcsv1jsonjsontable
age: 35 city: Seattle		Metrics: we check if the	prediction results exact match with the labels.

Figure 13: A bad case of format check task.



Figure 14: The pretrain stage of Amber-7B.

		Learning (Count &	x Navigation)	
Prompt: (n-shot exam	nples)		Prompt: (n-shot exam	uples)
Input: down, do Output: up: 0, d	wn own: 2		Input: left, left, Output: up: 0, d	right, left, down, left, right, right own: 1, left: 4, right: 3
Input: up In Output: up: 1, down: 0 C		Input: up, down Output: up: 2, d	Input: up, down, up, right, right, right, right, down Output: up: 2, down: 2, left: 0, right: 4	
Input: up, up, de Output:	own		Input: left, right Output:	, down, down, right, right, left, up, up, up
Tiny Llama:	up: 2, down: 1		Tiny Llama:	up: 3, down: 3, left: 3, right: 3
LLaMA2-7B:	up: 2, down: 1		LLaMA2-7B:	up: 0, down: 2, left: 2, right: 4
Mistral-7B:	up: 2, down: 1		Mistral-7B:	up: 3, down: 2, left: 3, right: 3
Yi-6B:	up: 2, down: 1		Yi-6B:	up: 3, down: 3, left: 3, right: 3
Baichuan2-7B:	up: 2, down: 1		Baichuan2-7B:	up: 4, down: 4, left: 4, right: 4
Qwen-7B:	up: 2, down: 1		Qwen-7B:	up: 3, down: 3, left: 3, right: 3
Yi-34B:	up: 2, down: 1		Yi-34B:	up: 3, down: 2, left: 2, right: 3
LLaMA-65B:	up: 2, down: 1		LLaMA-65B:	up: 3, down: 3, left: 3, right: 3
Label:	up: 2, down: 1		Label:	up: 3, down: 2, left: 2, right: 3
Metrics: we will check if the prediction dictionary string is exact match with the label dictionary string.				

Figure 15: The bad cases of Count & Navigation task.

Learning (Numbers' Rules)

Prompt:

Input: [3, 41, 7, 1, 11, 77, 9, 61, 4, 17] Output: [41, 1, 77, 61, 17]

Input: [36, 9, 1, 17, 29, 4, 93, 89] Output: [9, 17, 89]

Input: [51, 61, 0, 9, 89, 85] Output: [61, 9, 85]

Input: [83, 25, 95, 55, 46, 6, 2, 9] Output: [25, 55, 9]

Input: [3, 5, 6, 21, 15, 77, 8, 86, 1] Output: [5, 21, 77]

Input: [90, 31, 8, 2, 7, 24] Output: [31]

Input: [61, 94, 4, 3, 65, 9, 30, 11, 6, 7] Output: [3, 9, 11, 7]

Input: [9, 6, 2, 64, 15, 84] Output: []

Input: [2, 17, 5, 62, 4, 99, 0] Output: [17, 99]

Input: [8, 2, 27, 7, 37, 4, 36, 3, 62, 15] Output: [7, 3, 15]

Input: [9, 8, 0, 58, 21, 3] Output: [3]

Input: [6, 5, 0, 37, 52, 36] Output: [5, 37]

Input: [4, 9, 6, 1, 8, 16, 97] Output: [9, 1]

Input: [5, 1, 48, 8, 2, 98, 13] Output: [1]

Input: [5, 10, 96, 79, 1, 7, 9, 99] Output: [79, 7, 99]

Input: [54, 5, 1, 8, 85, 47] Output: [5, 47]

Input: [55, 63, 92, 99, 43, 2, 9] Output:

Mistral-7B: [92, 99, 43]

[92, 99, 43] LLaMA-65B: [63, 92, 9]

Label: [63, 99]

Human Score: 0.495

Rule: keep only odd elements with an even index

Metrics:

Yi-34B:

we check if the prediction results exact match with the labels.

Prompt:

Input: [3, 8, 7, 57, 4, 74, 49, 74, 78] Output: [8, 7, 57]

Input: [2, 4, 3, 2] Output: [4, 3]

Input: [5, 8, 5, 0, 60, 80, 1] Output: [8, 5, 0, 60, 80]

Input: [7, 9, 81, 2, 6, 1, 5, 6, 3] Output: [9, 81, 2, 6, 1, 5, 6]

Input: [1, 68, 7] Output: [68]

Input: [2, 4, 74, 15] Output: [4, 74]

Input: [8, 6, 5, 55, 76, 9, 50, 32, 66, 33] Output: [6, 5, 55, 76, 9, 50, 32, 66]

Input: [2, 3, 39, 60, 95, 0, 16, 0, 2, 2] Output: [3, 39]

Input: [3, 51, 21, 21, 2] Output: [51, 21, 21]

Input: [9, 0, 8, 5, 52, 2, 12, 68, 6, 22] Output: [0, 8, 5, 52, 2, 12, 68, 6, 22]

Input: [3, 99, 90, 49, 0, 6, 9, 34, 43] Output: [99, 90, 49]

Input: [1, 3, 6, 1, 83] Output: [3]

Input: [1, 5, 0, 0] Output: [5]

Input: [5, 6, 35, 3, 67, 41, 27, 82] Output: [6, 35, 3, 67, 41]

Input: [8, 45, 6, 18, 94, 0, 1, 2, 7, 34] Output: [45, 6, 18, 94, 0, 1, 2, 7]

Input: [4, 97, 9, 71, 6, 31, 2, 77, 27, 65] Output: [97, 9, 71, 6]

Input: [1, 7, 66, 6, 93, 4, 47] Output:

Mistral-7B: [7, 66, 6, 93, 4]

Yi-34B: [7, 66, 6, 93, 4]

LLaMA-65B: [7, 66, 6, 93, 4]

Label: [7]

Human Score: 0.116 **Rule:** elements 2 through N + 1, N = element 1

Figure 16: The bad cases of numbers' rules task.

Natural Language Context (String Completion)

The c6803ea75c870d0e is the second largest inland river in Gansu Province, China, and one of the three major inland river systems in the Hexi Corridor.

Water system.

The c6803ea75c870d0e system is composed of the main stream Changma River, ...

Hydrology.

The c6803ea75c870d0e has a total length of 861 kilometers, with a water-bearing section of about 770 kilometers in summer and 540 kilometers in winter, ...

Utilization.

In 1734, a 30-kilometer-long channel was built in the Caotan of Yinmachang to divert water for irrigation. In the c6803ea75c870d0e Basin, 5 reservoirs, including the Shuangta Reservoir, the Danghe Reservoir and the Chijinxia Reservoir, have a water storage capacity of 1 million cubic kilometers. Reservoir above the foot. The alluvial plain in the middle and lower reaches is an agricultural area, and the river runoff decreases year by year due to the increase of irrigation diversion. Historically, the c6803ea7

5c870d0e

Metrics: we will check if the prediction hash string is exact match with the label hash string.

Structured Context (Dictionary Search)

0aede0: Rangitoto College opened in 1956, with an initial roll of 180 Year 9 and 10 students (then known as Forms 3 and 4). 097944 : Arsamosata shrank to a medium-sized town and some of its population probably migrated to Harput, ... of the region. 063fb5 : In 1858 Justus went to Shwekyin to find appropriate locations for posting native preachers; ... and died on 31 March 1858. c5f3b8 : On August 15, a convention in Atlanta, Georgia, representing 17 presbyteries, ... the General Assembly's jurisdiction. 6aae02 : She began in 2010 during a period when Metcalfe experimented with doom metal textures and ...and abstract compositions. 5baa0d : Moreover, cloudy days are often warm and the precipitations are concentrated in the warmest months, ...and September. 79bbc3 : In this period, the band won several prizes and established records. 71df7d : In winning the race for a fifth consecutive year, she equalled the feat of Golden Miller, ...Gold Cups in the 1930s. 340b02 : Nevertheless, it has been suggested that regulation of CDC6 is one of several redundant mechanisms ... in eukaryotic cells. 5flcf9: Naming cities such the "joy of" or "happiness of" was a Orontid practice that recalled the Achaemenid royal discourse. 7a4e5b : It also provided the impetus for the launch of a new project and debut album for the pair, "Project One". **594c14**: The latter split over slavery in 1858. 912baa : Electric fish were again reported millennia later by ancient Greek, Roman and Arabic naturalists and physicians. 839b94 : For 10 years, Underhill was a master housewright for Colonial Williamsburg. **b2c766** : The Upper Kirby YMCA Extension is located in Upper Kirby 7a934e : In June 2022 the Oromo Liberation Front besieged the town and heavy gunfire was reported. ab7a16: To overcome Socrates' objection to his second definition of piety, Euthyphro amends his definition. 2ed45e : Euthyphro uses Zeus as evidence for his notions of piety while disregarding Uranus and Cronus, for example. a60813: Retinoic acid produced by dendritic cells induces upregulation of LPAM-1 on these cells. 1740b6 : During this time, members (particularly drummers) were constantly joining and leaving the band. 71df7d : In winning the race for a fifth consecutive year, she equalled the feat of Golden Miller, ...Gold Cups in the 1930s. Metrics: we will check if the prediction string is exact match with the label string. Structured Context (Dictionary Search) 87 🛛 487 ¥ 984 ★ 19 🖉 915 苯 691 ★ 64 苯 77 🗳 43 ★ 926 苯 76 ☆ 564 🖉 704 🖉 847 🖉 854 🗮 203 🖉 3 🗮 470 🎽 132 🖉 718 🖉 46 🖉 84 🕸 80 😤 : 35180 87 🛛 487 🛠 984 🛠 19 🖉 912 苯 691 ★ 64 🕊 77 🗅 43 🕊 926 🕊 76 🚖 564 🖉 704 🖉 847 🖉 854 🕊 203 🖉 3 🕊 470 🅊 132 🖉 718 🖉 46 🖉 84 🚖 80 😤 : 35183 87 🕽 487 * 987 * 19 🔄 915 * 691 * 64 * 77 🐌 43 * 926 * 76 🕸 564 🖉 704 🖉 847 🔅 84 * 203 🖉 3 * 470 * 132 🖉 768 🖉 46 🖉 84 🕸 80 🚈 : 35133 85 🖉 487 * 984 * 19 🖉 915 * 691 * 64 * 77 🖉 43 * 926 * 76 🕸 564 🖉 704 🖉 847 🖉 854 * 203 🖉 3 * 471 * 132 🖉 718 🖉 46 🖉 84 🕸 80 🖄 : 35179 87 🖉 487 * 984 * 19 🖉 925 * 691 * 64 * 77 🖉 43 * 926 * 76 🕸 564 🖉 704 🖉 847 🖉 854 * 203 🖉 3 * 470 * 132 🖉 718 🖉 46 🖉 84 🕸 80 🖄 : 35179 87 🛛 487 🛪 984 ★ 19 🖉 915 \star 691 ★ 64 ★ 77 🗅 43 ★ 926 ★ 76 🕁 564 🖉 704 🖉 847 🖉 854 ★ 203 🖉 3 🛪 470 ★ 132 🖉 778 🖉 46 🖉 84 🕁 80 左 : 35120

87 〕 487 ★ 984 ★ 14 〕 915 ★ 691 ★ 64 ★ 77 〕 43 ★ 926 ★ 76 ☆ 564 〕 704 〕 847 〕 854 ★ 203 〕 3 ★ 470 ★ 132 〕 718 〕 46 〕 84 ☆ 80 ∑ : 35175 87 〕 987 ★ 984 ★ 19 〕 915 ★ 691 ★ 64 ★ 77 〕 43 ★ 926 ★ 76 ☆ 564 〕 704 〕 847 〕 854 ★ 203 〕 3 ★ 470 ★ 132 〕 718 〕 46 〕 84 ☆ 80 ∑ : 34680 87 〕 487 ★ 984 ★ 19 〕 912 ★ 691 ★ 64 ★ 77 〕 43 ★ 926 ★ 76 ☆ 564 〕 704 〕 847 〕 854 ★ 203 〕 3 ★ 470 ★ 132 〕 718 〕 46 〕 84 ☆ 80 ∑ : 34680

35183

Metrics: we will check if the prediction number is exact match with the label number.

Figure 17: The tasks for copying ability evaluation. The prompt and label are in the yellow block and green block respectively. The metrics description is in the red block.

	Format Check	
Input: Index name age city I Ava Hill 31 Portland Output: table Input: Index,name,age,city 1,David Wilson,29,Boston	Input: person: age: 2 city: L Output {"name: {"name	Bentley Jackson 7 .os Angeles : yaml ": "David Wilson", "age": 29, "city": "Boston"}
Output: csv Input: (Landon Smith, age, 31) (Landon Smith, city, New Orleans) Output: tuple Input: <person></person>	Output Input: compar name: addres state: busine Output	: jsonl y: Crimson Opera House s: 789 Sunrise Street New York ss direction: opera performances :
<name>Matthew Martinez</name> <age>33</age> <city>Denver</city> Output: xml	yaml Metric: the labe	: we check if the prediction results exact match with ls.
	Format Cloning (CoT-An	swer)

... (n-shot examples)

Question: While playing with her friends in their school playground, Katelyn saw 50 fairies flying above the nearby forest. After about twenty minutes, one of her friends saw half as many fairies as Katelyn saw come from the east and join the fairies that were there. Ten minutes later, 30 fairies flew away. How many fairies are remaining?

Response: Katelyn's friend saw 1/2*50 = 25, half as many fairies as Katelyn initially saw The total number of fairies flying above the forest now is 50 + 25 = 75If thirty fairies then flew away, there were 75 - 30 = 45 fairies that remained.

So the answer is 45

Question: The girls are trying to raise money for a carnival. Kim raises \$320 more than Alexandra, who raises \$430, and Maryam raises \$400 more than Sarah, who raises \$300. How much money, in dollars, did they all raise in total? **Response:**

\nSo the answer is value

Metrics: We only evaluate the correctness of the format and do not consider the actual answer. The model will generate responses based on the provided examples, and our focus is solely on verifying whether the output format matches the expected label.

Format Cloning (Choice-Value)

... (n-shot examples)

Question: Train 'A' leaves Mumbai Central for Lucknow at 11 am, running at the speed of 40 kmph. Train 'B' leaves Mumbai Central for Lucknow by the same route at 2 pm on the same day, running at the speed of 72 kmph. At what time will the two trains meet each other?

Options: A)12 am on the next day, B)5 am on the next day, C)5 pm on the next day, D)2 pm on the next day, E)None of these **Answer:** <string> 12 am on the next day </string>

Question: You can rent DVDs at a local video store for \$4.00 per movie without a membership. However, if you purchase a membership for \$7.00 per month, you can rent DVDs for \$2.00 each. What is the minimum amount of DVDs you would have to rent to make it worth it to purchase the membership? **Options:** A)1, B)2, C)3, D)4, E)5

Answer:

<string> 2 </string>

Metrics : We only evaluate the correctness of the format and do not consider the actual answer. For instance, if the predicted result matches the format "<string>", "<string>", "<string>", ..., or "<string>", we consider it as correct. However, if the predicted result is in a different format, such as "<string> B </string>", "B", or "2", we consider it as incorrect.

Figure 18: The samples of format check and format cloning tasks. The prompt and label are in the yellow block and green block respectively. The metrics description is in the red block.

Format Conversion (Transfer)

... (n-shot examples)

Input:

|Index|name|age|city| |---|---| |1|Elijah Morgan|36|Pittsburgh| Output: <person>

- <name>Elijah Morgan</name> <age>36</age>
- <city>Pittsburgh</city>

</person>

Input:

|Index|name|address|state|business direction| |---|---|---|---| |1|Nebula Playhouse|789 Stardust Street|Utah|live performances| Output:

<company>

<name>Nebula Playhouse</name>

<address>789 Stardust Street</address>

<state>Utah</state>

Metrics : We evaluate the predictions based on an exact match with the labels.

Format Conversion (Multi)

... (n-shot examples)

Input:

Index,name,age,city 1,Aiden Taylor,33,Las Vegas 2,Carter Young,31,Salt Lake City 3,Leo Davis,32,Indianapolis 4,Violet Morgan,36,Pittsburgh **Output:** person: name: Aiden Taylor age: 33 city: Las Vegas person:

name: Carter Young age: 31 city: Salt Lake City

person: name: Leo Davis age: 32 city: Indianapolis

person: name: Violet Morgan age: 36 city: Pittsburgh

Input:

Index,name,age,city 1,Paisley Turner,33,Kansas City 2,Levi Johnson,32,Chicago 3,Henry Cook,34,Cleveland **Output:**

Index,name,age,city\n1,Brooklyn Wilson,35,Tampa\n

Metrics : We evaluate the predictions based on an exact match with the labels.

Format Conversion (Single)

... (n-shot examples)

Input: |Index|name|age|city| |---|---|---| |1|Landon Smith|31|New Orleans| Output: Index,name,age,city 1,Landon Smith,31,New Orleans

Input: |Index|name|age|city| |---|---|---| |1|Brooklyn Wilson|35|Tampa| Output:

Index,name,age,city\n1,Brooklyn Wilson,35,Tampa\n

Metrics : We evaluate the predictions based on an exact match with the labels.

Format Conversion (Mix)

... (n-shot examples) Input: (William Rodriguez, age, 36) (William Rodriguez, city, Phoenix) (Easton Mitchell, age, 34) (Easton Mitchell, city, Cleveland) (Samuel Powell, age, 36) (Samuel Powell, city, Pittsburgh) (Mason Lee, age, 29) (Mason Lee, city, San Francisco) (Dusk Dance Studio, address, 987 Twilight Lane) (Dusk Dance Studio, state, Florida) (Dusk Dance Studio, business direction, dance classes) (Starry Night Dance Company, address, 789 Stardust Street) (Starry Night Dance Company, state, Louisiana) (Starry Night Dance Company, business direction, dance performances) (Starshine Stadium, address, 456 Galaxy Lane) (Starshine Stadium, state, Florida) (Starshine Stadium, business direction, sports games) (Harmony Valley Vineyard, address, 321 Vineyard Lane) (Harmony Valley Vineyard, state, Oregon) (Harmony Valley Vineyard, business direction, winery) (Eclipse Theater, address, 456 Eclipse Street) (Eclipse Theater, state, Louisiana) (Eclipse Theater, business direction, live performances) **Output:** |Index|name|age|city|

|---|---|---| |1|William Rodriguez|36|Phoenix| |2|Easton Mitchell|34|Cleveland| |3|Samuel Powell|36|Pittsburgh| |4|Mason Lee|29|San Francisco|

|Index|name|address|state|business direction| |---|---|---|---| |1|Dusk Dance Studio|987 Twilight Lane|Florida|dance classes| |2|Starry Night Dance Company|789 Stardust Street|Louisiana|dance performances| |3|Starshine Stadium|456 Galaxy Lane|Florida|sports games| |4|Harmony Valley Vineyard|321 Vineyard Lane|Oregon|winery| |5|Eclipse Theater|456 Eclipse Street|Louisiana|live performances|

Metrics: we check if the prediction results exact match with the labels.

Figure 19: The samples of format conversion tasks with four different forms: "single", "multi", "transfer" and "mix".

Order Check	Specify Order (Word-Level)
(n-shot examples)	(n-shot examples)
Input: String1: guess, bid, record, cream, literature String2: guess, bid, record, cream, literature Output: False	Input: activity, brief, wonder, anger, leave Output: anger, wonder, activity, brief, leave Input: market, forever, will, curve, world Output: curve, will, market, forever, world
String1: elevator, juice, fall, hurt, spray String2: spray, hurt, fall, juice, elevator Output: True	Input: pain, leading, drag, shoot, medicine Output: shoot, drag, pain, leading, medicine
Input: String1: sound, wave, voice, weather, cake String2: cake, weather, voice, wave, sound Output:	Input: view, band, confidence, focus, mixture Output: focus, confidence, view, band, mixture Input: shopping, drama, care, start, passenger Output:
True	start, care, shopping, drama, passenger
Metrics: we check if the prediction results exact match with the labels.	Metrics: we evaluate the predictions based on an exact match with the labels.
Keep Order (Word-Level)	Reverse Order (Word-Level)
(n-shot examples)	(n-shot examples)
Input: wonder, annual, potential, landscape, purple, equal, style, department, diamond Output: wonder, annual, potential, landscape, purple, equal, style, department, diamond	Input: importance, god, quiet, hunt, present, train, building, pain, support, box, care, neck, high Output: high, neck, care, box, support, pain, building, train, present, hunt, quiet, god, importance
Input: square, spot, computer, historian, sample, great, pass, minimum, unit Output:	Input: brain, hotel, leave, other, assist, clerk, anybody, fly, bell, salary, gate, remove, stranger, convert Output:
square, spot, computer, historian, sample, great, pass, minimum, unit	convert, stranger, remove, gate, salary, bell, fly, anybody, clerk, assist, other, leave, hotel, brain
Metrics: we evaluate the predictions based on an exact match with the labels.	Metrics: we check if the prediction results exact match with the labels.
Keen Order (Sente	nce-Level)

... (n-shot examples)

Input: As such, "Nusayri" is now generally regarded as antiquated, and has even come to have insulting and abusive connotations. According to the "Caucasian calendar" of 1912, the village had 1,424 inhabitants, the majority of whom were Azerbaijanis, who were listed as "Tatars" in the calendar.

It was added to the Queensland Heritage Register on 24 March 2000. **Output:**

As such, "Nusayri" is now generally regarded as antiquated, and has even come to have insulting and abusive connotations. According to the "Caucasian calendar" of 1912, the village had 1,424 inhabitants, the majority of whom were Azerbaijanis, who were listed as "Tatars" in the calendar.

It was added to the Queensland Heritage Register on 24 March 2000.

Metrics: we will check if the prediction content is exact match with the label content.

Figure 20: The samples of order check and order adjustment tasks. The prompt and label are in the yellow block and green block respectively. The metrics description is in the red block.

Statistics (Count)

... (n-shot examples)

Input: left, up, up, down, down, up, left, up, down **Output:** up: 4, down: 3, left: 2, right: 0

Input: right, down, up, down, up, up, left, left **Output:** up: 3, down: 2, left: 2, right: 1

Input: left, down, right, down, up, up **Output:** up: 2, down: 2, left: 1, right: 1

Input: up, up, up, right, right, up **Output:**

up: 4, down: 0, left: 0, right: 2

Metrics: we check if the prediction results exact match with the labels.

Statistics (Relation Analysis)

... (n-shot examples)

Input:

umber_emu_588 and pink_baboon_823 are friends. violet_lobster_885 and pink_baboon_823 are friends. orange_alpaca_795 and umber_emu_588 are friends. mauve_squirrel_890 and pink_baboon_823 are friends. violet_lobster_885 and umber_emu_588 are friends. mauve_squirrel_890 and umber_emu_588 are friends. orange_alpaca_795 and pink_baboon_823 are friends. violet_lobster_885 and orange_alpaca_795 are friends. mauve_squirrel_890 and orange_alpaca_795 are friends. Query: Who are the friends of violet_lobster_885? Output:

orange_alpaca_795, umber_emu_588, pink_baboon_823

Metrics: we check if the prediction results exact match with the labels.

Statistics (Navigation)

... (n-shot examples)

Input: right, right, up, left, down, up, down, left, down **Output:** down: 1, left: 0

Input: left, down Output: down: 1, left: 1

Input: left, right, up, right, left, left **Output:** up: 1, left: 1

Input: left, right, up, right, left Output:

up: 1, left: 0

Metrics: we check if the prediction results exact match with the labels.

Statistics (Duplication Check)

... (n-shot examples)

Input: string, rate, judgment, tooth **Output:** False

Input: weather, virus, conversation, conversation Output: True

Input: wall, beer, enthusiasm, hook, revenue, revenue Output: True

Input: sound, champion, side, lift, sound **Output:**

True

Metrics: we check if the prediction results exact match with the labels.

Statistics (De-Duplication)

... (n-shot examples)

Input: Political information is obviously important here, but consumer preferences can play a role as well. As the pie charts show, in 1881, the most popular trades were agriculture, domestic services and working with animals. As the pie charts show, in 1881, the most popular trades were agriculture, domestic services and working with animals. As the pie charts show, in 1881, the most popular trades were agriculture, domestic services and working with animals.

It is the twelfth track on their fourteenth studio album, "Songs of Experience", and was released as its third single on 23 April 2018.

Flora includes sugar cane and tamarind, which are typically served as local dishes. "The Woodwright's Shop" teaches the art of traditional woodworking, using hand tools and human-powered machines.

As the pie charts show, in 1881, the most popular trades were agriculture, domestic services and working with animals.

Metrics: we will check if the prediction content is exact match with the label content.

Figure 21: The samples of tasks about statistics problem. The prompt and label are in the yellow block and green block respectively. The metrics description is in the red block.

Output:

Max Min	LLaMA-65B	LLaMA-30B	Yi-34B	Baichuan2-13B-Chat	Baichuan2-13B	Baichuan2-7B-Chat	Baichuan2-7B	Qwen-14B-Chat	Qwen-14B	Qwen-7B-Chat	Qwen-7B	Skywork-13B	Mistral-7B-Chat	Mistral-7B	InternLM-7B-Chat	InternLM-7B	LLaMA2-13B-Chat	LLaMA2-13B	LLaMA2-7B-Chat	LLaMA2-7B	LLaMA-13B	LLaMA-7B	Yi-6B-Chat	Yi-6B	ChatGLM3-6B-Chat	ChatGLM3-6B	GPT-J	GPT-Neo-1.3B	Phi-1.5	TinyLlama-1.1B	Random	Models
0.900 0.530	0.890	0.890	0.860	0.750	0.800	0.880	0.870	0.900	0.870	0.890	0.890	0.880	0.870	0.880	0.530	0.550	0.900	0.890	0.890	0.890	0.890	0.870	0.870	0.870	0.890	0.900	0.530	0.540	0.540	0.890	0.000	String Completion
0.763 0.232	0.495	0.332	0.600	0.363	0.232	0.516	0.442	0.584	0.484	0.721	0.453	0.458	0.763	0.537	0.732	0.421	0.747	0.474	0.758	0.453	0.426	0.332	0.484	0.379	0.342	0.700	0.463	0.479	0.253	0.458	0.000	Dictionary Search
0.675 0.000	0.000	0.475	0.117	0.575	0.633	0.058	0.158	0.167	0.308	0.083	0.100	0.492	0.450	0.217	0.575	0.317	0.133	0.117	0.258	0.158	0.192	0.358	0.033	0.100	0.675	0.483	0.617	0.175	0.175	0.375	0.167	Format Check
0.960 0.427	0.947	0.940	0.960	0.887	0.893	0.773	0.933	0.546	0.923	0.813	0.927	0.953	0.917	0.947	0.427	0.877	0.900	0.947	0.920	0.917	0.907	0.900	0.863	0.943	0.887	0.840	0.833	0.777	0.720	0.827	0.000	Format Cloning
0.842 0.342	0.833	0.783	0.808	0.683	0.725	0.642	0.800	0.758	0.808	0.342	0.700	0.808	0.833	0.842	0.425	0.767	0.658	0.750	0.650	0.767	0.800	0.775	0.808	0.792	0.725	0.475	0.675	0.492	0.550	0.592	0.000	Format Conversion
0.980 0.540	0.970	0.910	0.980	0.830	0.870	0.670	0.610	0.770	0.870	0.630	0.750	0.930	0.720	0.980	0.640	0.630	0.750	0.830	0.730	0.830	0.650	0.670	0.640	0.740	0.610	0.750	0.730	0.610	0.550	0.540	0.500	Order Check
0.729 0.308	0.729	0.629	0.696	0.396	0.450	0.429	0.454	0.683	0.717	0.354	0.487	0.438	0.471	0.575	0.308	0.404	0.604	0.608	0.454	0.429	0.442	0.412	0.525	0.488	0.363	0.333	0.438	0.387	0.379	0.379	0.000	Order Adjustment
0.730 0.500	0.613	0.637	0.730	0.630	0.617	0.557	0.603	0.720	0.707	0.643	0.687	0.657	0.603	0.623	0.620	0.577	0.620	0.620	0.623	0.630	0.577	0.587	0.623	0.500	0.550	0.517	0.637	0.533	0.573	0.557	0.500	Duplication Check
0.637 0.000	0.563	0.447	0.627	0.510	0.467	0.403	0.337	0.623	0.637	0.337	0.523	0.357	0.493	0.490	0.000	0.277	0.467	0.423	0.403	0.293	0.340	0.327	0.447	0.340	0.327	0.273	0.233	0.180	0.287	0.223	0.000	De- Duplication
0.470 0.000	0.330	0.350	0.470	0.270	0.210	0.140	0.110	0.350	0.350	0.110	0.150	0.130	0.190	0.300	0.000	0.100	0.260	0.090	0.220	0.120	0.080	0.030	0.140	0.070	0.100	0.170	0.030	0.010	0.030	0.010	0.000	Relation Analysis
0.525 0.092	0.392	0.425	0.525	0.392	0.325	0.233	0.308	0.467	0.467	0.300	0.333	0.350	0.358	0.442	0.258	0.158	0.292	0.308	0.317	0.233	0.275	0.150	0.258	0.217	0.225	0.317	0.133	0.092	0.100	0.125	0.000	Count & Navigation
0.576 0.356	0.576	0.528	0.536	0.356	0.392	0.456	0.492	0.552	0.552	0.492	0.484	0.472	0.468	0.560	0.380	0.456	0.488	0.520	0.412	0.452	0.472	0.416	0.452	0.492	0.372	0.432	0.492	0.376	0.388	0.356	0.000	Complex rules
0.675 0.396	0.646	0.619	0.675	0.541	0.540	0.538	0.551	0.632	0.653	0.559	0.574	0.603	0.640	0.648	0.468	0.479	0.631	0.595	0.609	0.560	0.547	0.518	0.554	0.541	0.526	0.582	0.506	0.430	0.396	0.498	0.072	Mean

Table 6: The detailed results for ICLEval of all evaluated models.