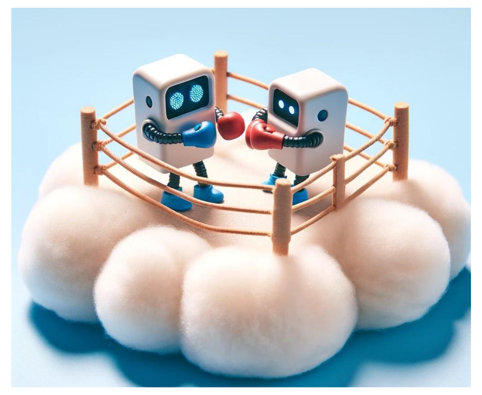
CloudEval-YAML: A Realistic and Scalable Benchmark for Cloud Configuration Generation

Yifei Xu*, Yuning Chen*, Xumiao Zhang*, Xianshang Lin, Pan Hu†, Yunfei Ma, Songwu Lu, Wan Du, Z. Morley Mao, Ennan Zhai, Dennis Cai

*Equal Contribution †Corresponding Author

Introduction

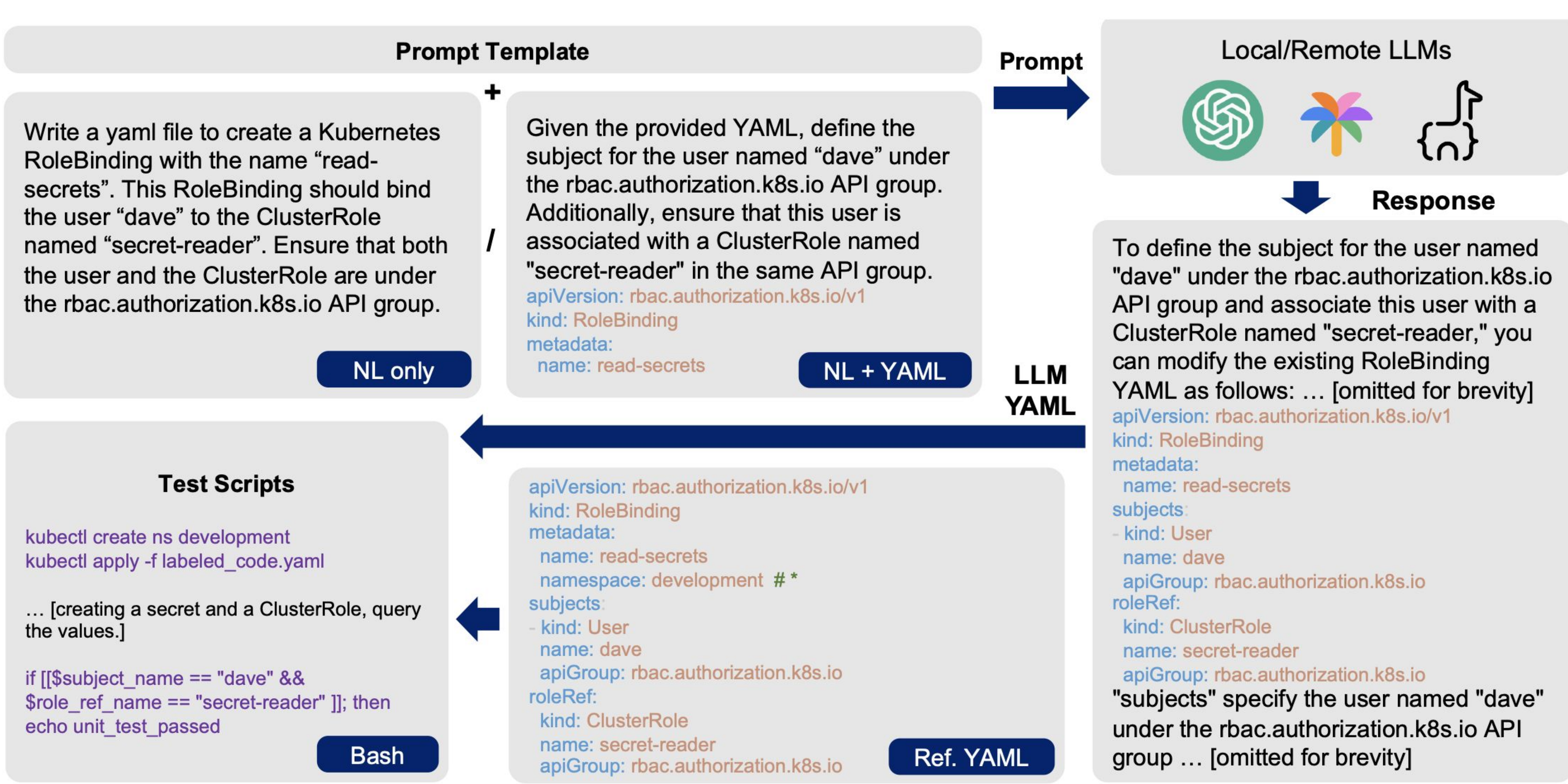


- We present `CloudEval-YAML`, a first benchmark for LLM in generating config for cloud applications, which includes handwritten dataset with 337 original problems, and 1011 total problems with abbreviated and bilingual augmentation.
- We present the design of a scalable, automated evaluation platform consisting of a computing cluster to evaluate the generated code efficiently for various performance metrics.
- We present an in-depth evaluation of 13 LLMs with `CloudEval-YAML`, including GPT-4, PaLM 2 and Llama 2, and show some preliminary findings

Dataset

Overall Structure

- **Problem Template:** Providing context for instruction-based LLMs, as well as specifying the output format
- **Natural Language Problems:** NL only or NL with YAML context
- **Reference YAML with Labels:** Correct solutions to the problems with labels in comments indicating non-critical fields
- **Unit Test Scripts:** Benchmarking functional correctness of the generated YAML



Problem Statistics

- **Applications**: 337 carefully constructed original problems targeting Cloud Applications including Kubernetes, Envoy, and Istio
- **Topics**: hand-picked from official documentation websites, popular issues from StackOverflow, and highly-ranked blog posts

Statistics	Kubernetes						Envoy	Istio	Total / Avg.
	pod	daemonset	service	job	deployment	others			/ Max
Total Problem Count	48	55	20	19	19	122	41	13	337
Avg. Question Words	77.06	80.91	71.35	73.74	94.84	69.48	275.56	73.00	99.40
Avg. Lines of Solution	18.67	23.58	15.00	20.37	29.00	19.74	85.85	14.92	28.35
Avg. Tokens of Solution	64.02	71.91	41.40	74.53	79.42	58.78	242.34	39.54	84.28
Max Tokens of Solution	150	111	83	163	140	194	531	53	531
Avg. Lines of Unit Test	8.52	8.58	11.25	7.68	12.53	17.74	11.56	20.00	13.14

Data Augmentation

According to a survey of Alibaba's cloud operation team, we augment the data with 2 types of questions derived from the original questions:

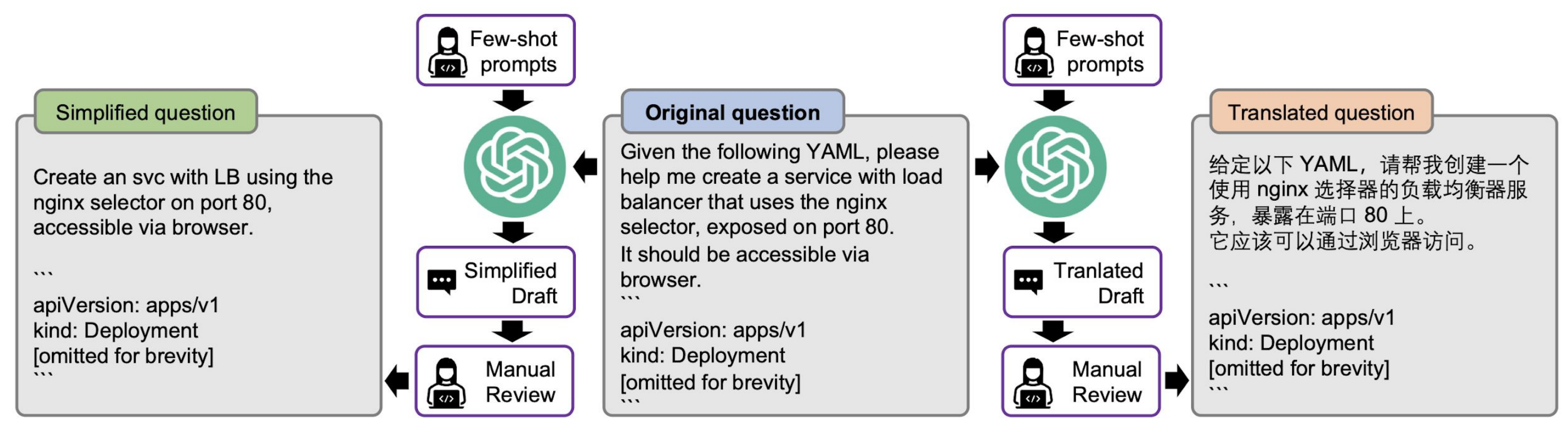
- **Simplified Question:** Short and clear language with domain-specific abbreviations
- **Translated Question:** Daily language used by Chinese cloud operation teams

Methodology

- We use GPT-4 [1] and few-shot prompting to generate simplified and translated drafts from original questions
- We manually review all drafts to ensure quality

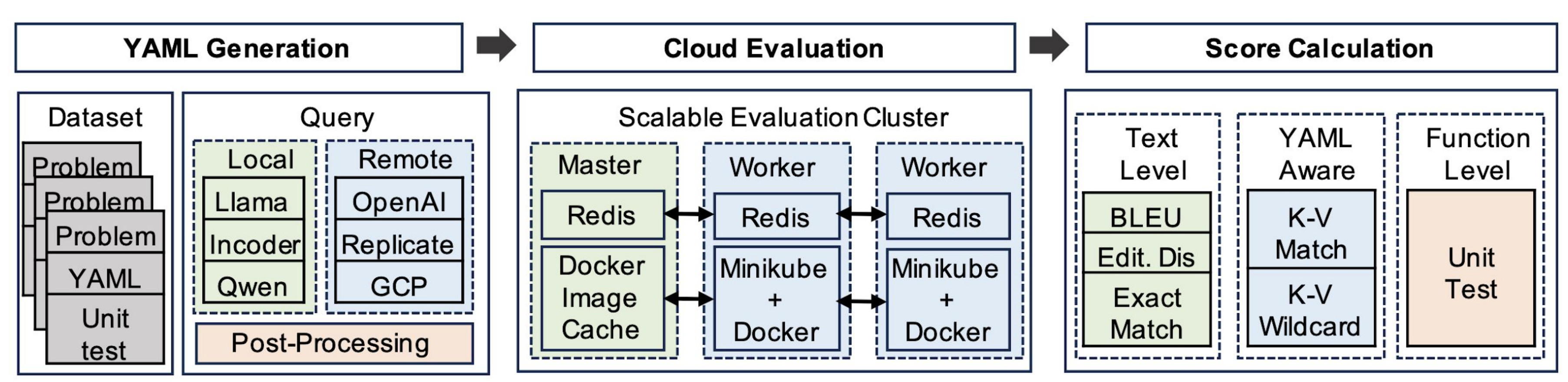
Statistics of Augmented Dataset

	Original	Simplified	Translated
Count	337	337	337
Avg. words	99.40	73.86 (-25.7%)	57.18
Avg. tokens	508.9	402.5 (-20.9%)	378.5



Benchmark Platform

Overall Workflow



Evaluation Metrics

- **BLEU:** Common metric used to evaluate the quality of machine-generated translations
- **Edit Distance:** The number of lines to edit between the generated YAML and the reference YAML
- **Exact Match:** Whether the generated YAML is identical to the reference YAML
- **K-V Exact Match:** Whether the generated and reference YAML are equivalent under YAML semantics
- **K-V Wildcard Match:** Similar to K-V Exact Match but with flexibility according to the labeled non-critical fields
- **Unit Test:** Whether the generated YAML can functionally fulfill the need of the question

(All metrics are normalized to [0, 1], the higher the better)

Optimizations for Evaluation Speed

- **Parallel Query:** We use `ray` [2] to parallelize the query for remote LLMs like GPT
- **Evaluation Cluster:** We support cluster-based evaluation to run unit tests on multiple machines in parallel, speeding up the process by over 20x

Evaluation Results

Overall Scores of 13 LLMs

Ranking	Model			Text-level Score			YAML-Aware Score		Function-level Score
	Name	Size	Open Source	BLEU	Edit Dist.	Exact Match	Key-value Exact	Key-value Wildcard	Unit Test ↓
1	GPT-4 Turbo	?	N	**0.649**	**0.551**	**0.099**	**0.208**	**0.667**	**0.561**
2	GPT-4	?	N	0.629	0.538	0.092	0.198	0.641	0.515
3	GPT-3.5	?	N	0.612	0.511	0.075	0.154	0.601	0.412
4	PaLM-2-bison [1]	?	N	0.537	0.432	0.040	0.092	0.506	0.322
5	Llama-2-70b-chat	70B	Y	**0.355**	**0.305**	0.000	**0.020**	**0.276**	**0.085**
6	Llama-2-13b-chat	13B	Y	0.341	0.298	0.000	0.016	0.265	0.067
7	Wizardcoder-34b-v1.0	34B	Y	0.238	0.247	**0.007**	0.013	0.230	0.056
8	Llama-2-7b-chat	7B	Y	0.289	0.231	0.000	0.009	0.177	0.027
9	Wizardcoder-15b-v1.0	15B	Y	0.217	0.255	0.002	0.002	0.226	0.026
10	Llama-7b	7B	Y	0.106	0.058	0.004	0.005	0.069	0.023
11	Llama-13b-lora	13B	Y	0.101	0.054	0.001	0.003	0.065	0.021
12	Codellama-7b-instruct	7B	Y	0.154	0.174	0.001	0.001	0.124	0.015
13	Codellama-13b-instruct	13B	Y	0.179	0.206	0.002	0.002	0.142	0.012

[1] The PaLM API supports English only at the time of submission so we averaged the score excluding translated questions.

- Proprietary models such as GPT-4 [1] are way ahead across all metrics, and the gap between them and the best performing open-source models is larger than in similar benchmarks like HumanEval [3]
- Code-specific LLMs typically perform poorly compared to general LLMs with similar or even smaller sizes in terms of the Unit Test score

Performance across Different Question Types

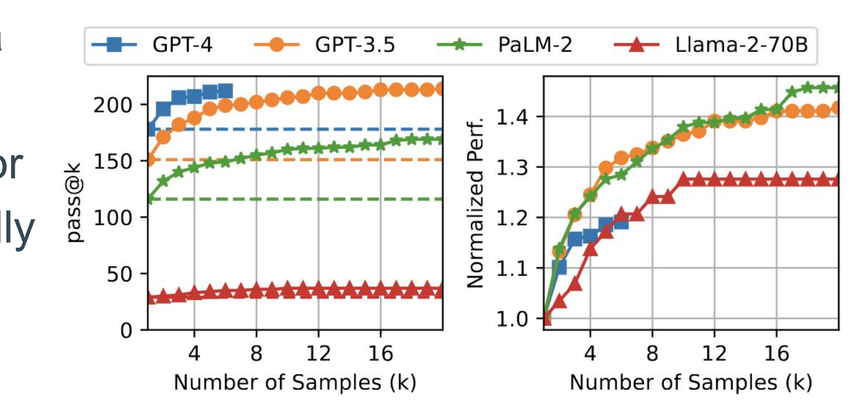
- Simplification of problems generally leads to lower performance, but larger models tends to be more resilient
- Code-specific and small models are severely affected by translation, while larger models keep up their performance relatively well

Model	Data Set		
Name	Original	Simplified	Translated
GPT-4	179	164 (-15)	178 (-1)
GPT-3.5	142	143 (+1)	132 (-10)
PaLM-2-bison	120	97 (-23)	N/A [1]
Llama-2-70b-chat	30	24 (-6)	32 (+2)
Llama-2-13b-chat	26	17 (-9)	25 (-1)
Wizardcoder-34b-v1.0	24	31 (+7)	2 (-22)
Llama-2-7b-chat	13	9 (-4)	5 (-8)
Wizardcoder-15b-v1.0	12	11 (-1)	3 (-9)
Llama-7b	12	7 (-5)	4 (-8)
Llama-13b-lora	8	9 (+1)	4 (-4)
Codellama-7b-instruct	5	6 (+1)	4 (-1)
Codellama-13b-instruct	5	2 (-3)	5 (+0)

Unit Test Scores on Different Question Types

Multi-sample Generation

- Multi-sample generation could be a good choice to improve the performance if there is a unit test for verification, or the user can manually select the best result.
- It can be cost-efficient to use a weaker-yet-cheaper model with multiple samples to outperform stronger ones.



Unit Test Scores of Multi-sample Generation (on the Original Dataset)

References

[1] GPT-4. https://openai.com/gpt-4 , 2023.

[2] Moritz, P., et al. Ray: A distributed framework for emerging {AI} applications. In 13th USENIX symposium on operating systems design and implementation (OSDI 18), pp. 561–577, 2018.

[3] Mark Chen, et al. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374, 2021.

Github Link