Projet de recherche

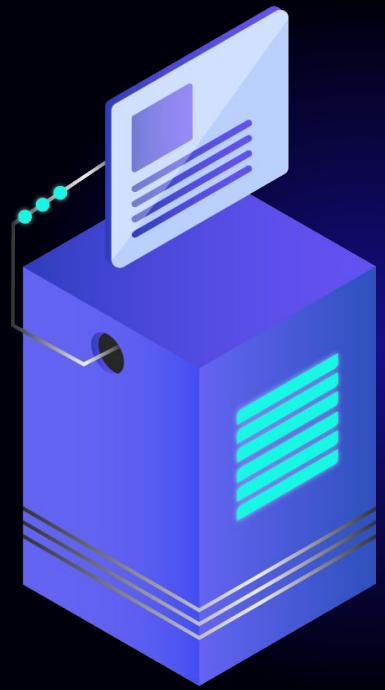
LLM 4 Cloud

Évaluation des capacités des LLM dans la génération de fichiers de configuration pour le Cloud

Adam OUZEGDOUH • Pierre LAURENT
Supervision : Hayet BRABRA • Walid GAALOUL

TELECOM SudParis

INSTITUT POLYTECHNIQUE DE PARIS



INTRODUCTION

Ce projet a deux objectifs principaux* :

- **Evaluer** les capacités des LLM à générer des fichiers de configuration pour le Cloud
- Puis explorer des approches visant à **améliorer** les résultats des LLM

**Les objectifs ont évolué après l'étude de l'état de l'art de la recherche sur le sujet.*

GROUPE DU PROJET

Etudiant

Adam OUZEGDOUH

Télécom SudParis • ENSIIE
Voie d'approfondissement DSI

Etudiant

Pierre LAURENT

Télécom SudParis • ENSIIE
Voie d'approfondissement DSI

Encadrant

Walid GAALOUL

Télécom SudParis *(Palaiseau)*
Professeur • Responsable de l'équipe de recherche ACMES

Encadrante

Hayet BRABRA

Télécom SudParis *(Palaiseau)*
Ingénieure de recherche

00

CONTEXTE

Contexte du projet

CONTEXTE DU PROJET

Voici ce que nous allons aborder :

- Qu'est-ce qu'un LLM ?
- Quelle est la signification des termes EC2, S3 ou encore RDS dans le lexique du monde cloud ?
- Qu'est-ce qu'un fichier de configuration Cloud ?
- Les fichiers d' Infrastructure as Code (IaC) / Couche infrastructure
- Les fichiers d'orchestration d'applications / Couche logicielle

CONTEXTE DU PROJET

Qu'est-ce qu'un LLM ?

Un **LLM (Large Language Model)** est un modèle d'intelligence artificielle entraîné sur d'énormes quantités de texte pour comprendre et générer du langage naturel.
Il fonctionne en prédictant le token* suivant le plus probable dans une séquence donnée.
**Un token est une unité de texte (mot, partie de mot ou ponctuation).*

Comment ça marche :
À partir d'un texte d'entrée, le modèle calcule les probabilités des tokens pouvant suivre et choisit le plus probable. Puis, il répète ce processus pour générer une phrase complète.

Exemple :
Pour "Le ciel est", le modèle prédit "bleu" car c'est le token le plus probable dans ce contexte.

Applications :
Traduction, résumé de documents, **génération de code informatique**, correction/reformulation de texte, rédaction d'articles, génération d'histoires ou de scripts, etc...

CONTEXTE DU PROJET

Quelle est la signification des termes EC2, S3 ou encore RDS dans le lexique du monde cloud ?

Elastic Compute Cloud (EC2) :

Service de machines virtuelles d'Amazon Web Services.

Équivalents : ***Machines virtuelles Azure*** *pour Microsoft Azure et* ***Compute Engine*** *pour Google Cloud Platform*

Simple Storage Service (S3) :

Service de stockage de fichiers et données d'Amazon Web Services.

Équivalents : ***Stockage Blob Azure*** *pour Microsoft Azure et* ***Cloud Storage*** *pour Google Cloud Platform*

Relational Database Service (RDS) :

Service de bases de données relationnelles d'Amazon Web Services.

Équivalents : ***Azure SQL*** *pour Microsoft Azure et* ***Cloud SQL*** *pour Google Cloud Platform*

CONTEXTE DU PROJET

Qu'est-ce qu'un fichier de configuration Cloud ?

Un **fichier de configuration pour le Cloud** est un fichier qui définit les paramètres et les ressources nécessaires pour déployer, configurer ou gérer des infrastructures et services dans un environnement cloud.

Ces fichiers permettent d'automatiser la gestion des ressources cloud tout en assurant la reproductibilité des déploiements.

Il existe deux types principaux d'usages pour les fichiers de configuration pour le cloud :
L' Infrastructure as Code (IaC) et l'orchestration d'applications.

Les deux usages sont complémentaires :
L' Infrastructure as Code prépare l'infrastructure, et l'orchestration déploie les applications.

CONTEXTE DU PROJET

Les fichiers d' Infrastructure as Code (IaC) :

Objectif :
Créer et provisionner des **ressources** cloud (serveurs, bases de données, etc...) sous forme de code.

Exemples d'outils : Terraform (.tf) mais aussi AWS CloudFormation ou Azure Resource Manager...

main.tf (Terraform) *version simplifiée*

```
# Spécification du cloud provider
provider "aws" {
    region = "eu-west-3" # Europe (Paris)
}

# Ressource, création EC2
resource "aws_instance" "example" {
    ami = "ami-xxxxxxxxxxxxx" # Amazon Machine Image (ex Debian 12)
    instance_type = "t2.nano" # instance avec 1 vCPU et 0.5 Gb de RAM
}
```

La CLI Terraform

terraform init : Initialise le projet et télécharge les plugins.

terraform plan : Prévisualiser les changements.

terraform apply : Applique la configuration et crée les ressources.

terraform destroy : Supprime toutes les ressources provisionnées.

CONTEXTE DU PROJET

Les fichiers d'orchestration d'applications :

Qu'est-ce que l'orchestration ?
L'orchestration est un processus automatisé permettant de coordonner et de gérer des workflows complexes composés de plusieurs composants logiciels. Elle permet :

- L'automatisation des déploiements
- La gestion de la scalabilité (montée ou descente en charge)
- Le suivi de la disponibilité des services.

Kubernetes est un outil d'orchestration qui propose différents services :

- **ClusterIP :** Expose le service uniquement à l'intérieur du cluster
- **NodePort :** Rend le service accessible à partir d'un port spécifique de chaque nœud
- **LoadBalancer :** Crée un équilibreur de charge externe pour distribuer le trafic
- **Ingress :** Gère le routage du trafic HTTP/HTTPS vers les services internes

La configuration des objets Kubernetes se fait via des fichiers **YAML** qui permettent de décrire de manière déclarative les ressources (**pods, services, déploiements, etc.**) pour :

- Spécifier l'état désiré des applications.
- Automatiser la gestion et le provisionnement des ressources.
- S'assurer que les configurations peuvent être versionnées et partagées facilement.

CONTEXTE DU PROJET

Les fichiers d'orchestration d'applications :

Cas d'application :
L'objectif ici est de créer un *RoleBinding* dans Kubernetes, qui lie un utilisateur nommé '*dave*' à un rôle appelé '*secret-reader*'. Ce rôle appartient au groupe d'API *rbac.authorization.k8s.io* et permet d'accorder des permissions spécifiques à cet utilisateur pour accéder aux ressources dans un cluster.

Exemple YAML

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: secret-reader-binding
  namespace: default
subjects:
- kind: User
  name: dave
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: secret-reader
  apiGroup: rbac.authorization.k8s.io
```

Explications

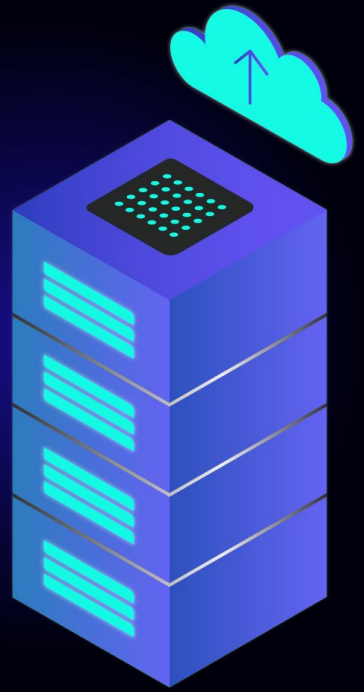
- **apiVersion :** Version de l'API utilisée pour le RBAC (Role-Based Access Control)
- **kind :** Type de ressource à créer (RoleBinding)
- **metadata :** Informations sur l'objet (nom et namespace)
- **subjects :** Définit l'utilisateur cible (dave)
- **roleRef :** Référence le rôle existant (secret-reader)

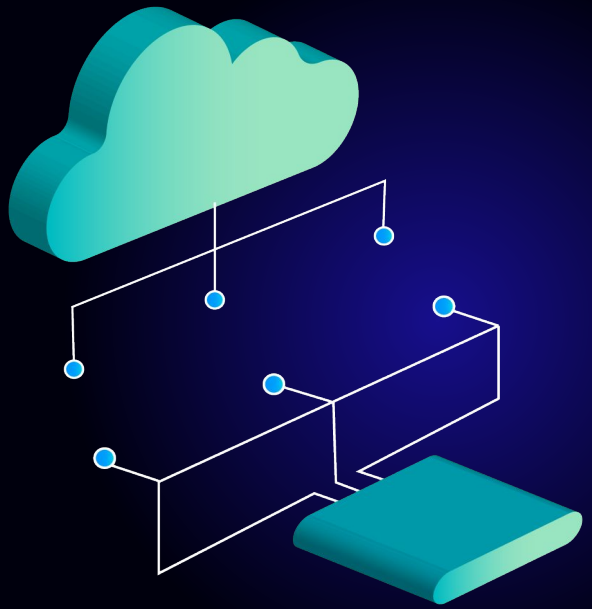
CONTEXTE DU PROJET

Scénarios cloud

DEFI N°1

Afin de tester les capacités des LLM à générer des fichiers de configuration pour le Cloud, nous avons besoin d'un vaste ensemble de scénarios (problématiques), ainsi que d'un mécanisme permettant d'évaluer et de valider la justesse des configurations produites.





CONTEXTE DU PROJET

Limiter les coûts

DEFI N°2

Le projet combine l'usage de ressources cloud et de LLM. Le défi des coûts est double, car il implique deux types de dépenses importantes :

- Les coûts liés aux ressources cloud
- Les coûts associés à l'utilisation des API des LLM (hors LLM local)

LES ETAPES DU PROJET DE RECHERCE

01 État de l'art

Recherches, puis étude de travaux existants

02 Ajustement

Évolution des objectifs du projet à la lumière des travaux existants

03? Benchmarking

Développement d'un outil d'évaluation des capacités des LLM

04? Stratégies

Expérimentation de stratégies pour améliorer les résultats

01

ETAT DE L'ART

Recherches, puis étude de travaux existants

ETAT DE L'ART

UNIVERSITY OF MICHIGAN

Sep. 2024

IaC-Eval: A Code Generation Benchmark for Infrastructure as Code Programs

OpenAI

Juil. 2021

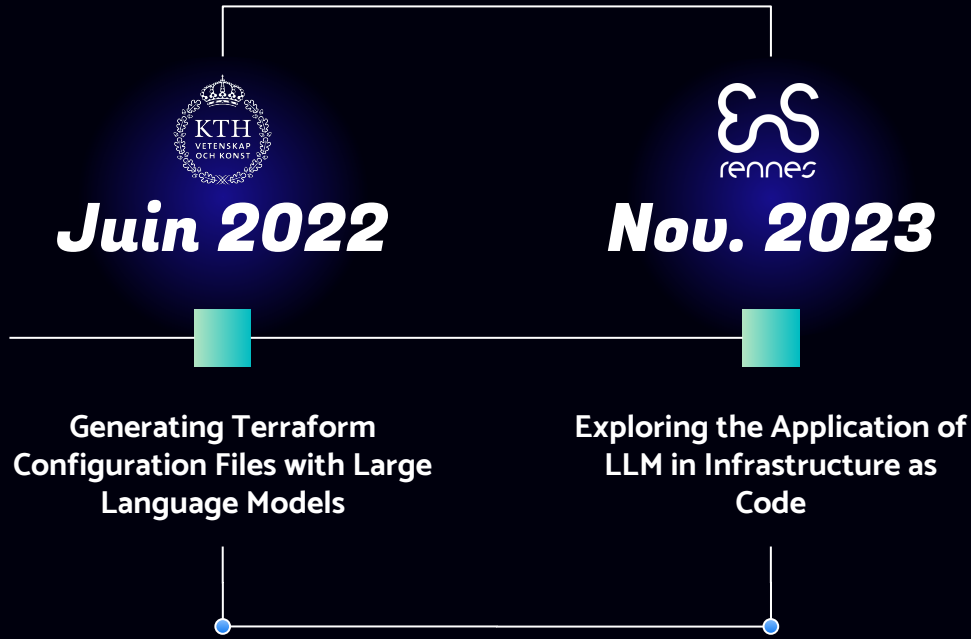
Evaluating Large Language Models Trained on Code

Alibaba Cloud

Nov. 2023

CloudEval-YAML: A Realistic and Scalable Benchmark

ETAT DE L'ART


KTH
VETENSKAP
OCH KONST
rennes
Juin 2022
Nov. 2023
Generating Terraform Configuration Files with Large Language Models
Exploring the Application of LLM in Infrastructure as Code

TELECOM SudParis

IP PARIS

Sep. 2024

IaC-Eval: A Code Generation Benchmark for Infrastructure as Code Programs

IaC-Eval: A Code Generation Benchmark for Infrastructure as Code Programs

Sep. 2024 • ***Patrick Tser Jern Kon, Jiachen Liu, Yiming Qiu, Weijun Fan, Ting He Lei Lin, Haoran Zhang, Owen M. Park, George S. Elengikal, Yuxin Kang Ang Chen, Mosharaf Chowdhury, Myungjin Lee (Cisco), Xinyu Wang***

NeurIPS 2024
Rank A* ICORE

Aperçu : IaC-Eval propose un benchmark unique pour tester la capacité des LLM à générer du code Infrastructure-as-Code pour Terraform. Malgré des performances prometteuses dans des tâches simples, les LLM montrent **d'importantes limites face à des scénarios complexes**.

LLM testés : GPT (4, 3.5-turbo), **WizardCoder** (33B-V1.1), **Magicoder** (S-CL-7B), **CodeLlama** (7B, 13B, 34B), **Gemini** (1.0 Pro)

LLM évalués sur : Terraform (problématiques AWS)

Scénarios : 458 scénarios IaC AWS (créés et validés par des humains) :

- **Resource :** *ex : "aws_ami"*
- **Prompt :** *ex : "Create the latest Amazon Linux 2 AMI"*
- **Rego intent (OPA Rego) :** *ex : "package terraform default ami_latest.....constant_value[_] == "amazon" }"*
- **Difficulty :** *ex : 1 sur 5*
- **Reference output :** *ex : "data "aws_ami" "latest_amazon_linux_2" { mo.....values = ["amzn2-ami-hvm-*-x86_64-gp2"] } }"*
- **Intent :** *ex : "create aws_ami resource with name = any"*

IaC-Eval: A Code Generation Benchmark for Infrastructure as Code Programs

Sep. 2024 • *Patrick Tser Jern Kon, Jiachen Liu, Yiming Qiu, Weijun Fan, Ting He Lei Lin, Haoran Zhang, Owen M. Park, George S. Elengikal, Yuxin Kang Ang Chen, Mosharaf Chowdhury, Myungjin Lee (Cisco), Xinyu Wang*

NeurIPS 2024
Rank A ICORE*

Exemple de scénario :

Prompt: “Create the latest Amazon Linux 2 AMI”

Reference output:

```
data ”aws_ami” ”latest_amazon_linux_2” {
 most_recent = true
 owners      = [”amazon”]

 filter {
  name   = ”name”
  values = [”amzn2-ami-hvm-*-x86_64-gp2”]
 }
}
```

Rego intent:

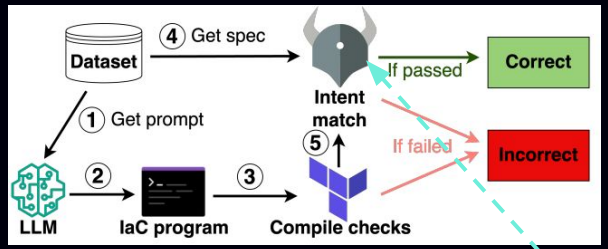
```
package terraform

default ami_latest_amazon_linux_2 = false

ami_latest_amazon_linux_2 {
 resource := input.configuration.root_module.resources[_]

 resource.type == ”aws_ami”
 resource.name == ”latest_amazon_linux_2”

 resource.expressions.filter[_].name.constant_value == ”name”
 resource.expressions.filter[_].values.constant_value[_] == ”amzn2-ami-hvm-*-x86_64-gp2”

 resource.expressions.most_recent.constant_value == true
 resource.expressions.owners.constant_value[_] == ”amazon”
}
```

IaC-Eval: A Code Generation Benchmark for Infrastructure as Code Programs

Sep. 2024 • *Patrick Tser Jern Kon, Jiachen Liu, Yiming Qiu, Weijun Fan, Ting He Lei Lin, Haoran Zhang, Owen M. Park, George S. Elengikal, Yuxin Kang Ang Chen, Mosharaf Chowdhury, Myungjin Lee (Cisco), Xinyu Wang*

NeurIPS 2024
Rank A ICORE*

Evaluation workflow



Scores

Rank	Model Name	Evaluation metric BLEU	CodeBERTScore	LLM-judge	IaC-Eval
1	GPT-4	18.49	83.39	61.79	19.36
2	WizardCoder-33B-V1.1	15.22	80.50	28.72	8.93
3	GPT-3.5-turbo	14.52	77.26	34.49	7.99
4	Magicoder-S-CL-7B	14.22	79.49	23.14	7.62
5	Gemini 1.0 Pro	11.96	78.90	19.72	3.43
6	CodeLlama Instruct (34B)	11.47	78.64	11.97	2.99
7	CodeLlama Instruct (13B)	11.18	76.46	9.83	2.01
8	CodeLlama Instruct (7B)	9.31	70.22	7.18	1.97

Terraform pour la compilation syntaxique
OPA Rego pour la validation de l'intention
Cela permet une évaluation sans nécessité de déploiement cloud !

IaC-Eval: A Code Generation Benchmark for Infrastructure as Code Programs

Sep. 2024 • **Patrick Tser Jern Kon, Jiachen Liu, Yiming Qiu, Weijun Fan, Ting He Lei Lin, Haoran Zhang, Owen M. Park, George S. Elengikal, Yuxin Kang Ang Chen, Mosharaf Chowdhury, Myungjin Lee (Cisco), Xinyu Wang**

NeurIPS 2024
Rank A* ICORE

Stratégies d'amélioration :

- **Stratégies des exemples (Few-shot) :** *Donner 3 exemples de problèmes + solutions en début de Prompt*
- **Etape par étape (CoT) :** *Donner 3 exemples de raisonnement étape par étape (First,.....Second,.....Finally,.....) en début de Prompt*
- **Boucle de prompts correctif (Multi-turn) :** *Lorsqu'une erreur est détectée, elle est renvoyée au LLM (Max 2 tours)*
- **Base de connaissances (RAG) :** *Le LLM identifie les termes-clés du problème pour récupérer les documents pertinents, puis les documents sont ensuite ajoutés au contexte de génération.*

Model		Enhancement strategy			
Rank	Name	Few-shot	CoT	Multi-turn	RAG
1	GPT-4	10.64	9.31	31.12	36.70
2	GPT-3.5-turbo	0.80	1.60	11.44	21.81
3	Magicoder-S-CL-7B	2.93	0.53	12.50	12.77
4	WizardCoder-33B-V1.1	1.60	1.06	9.04	11.70
5	CodeLlama Instruct (34B)	3.19	3.19	2.13	6.12
6	CodeLlama Instruct (7B)	2.39	3.72	0.53	5.59
7	Gemini 1.0 Pro	1.33	0.00	2.93	5.32
8	CodeLlama Instruct (13B)	1.06	1.86	1.06	3.46

GPT-4 obtenait un score de 19,36% sans amélioration

IaC-Eval: A Code Generation Benchmark for Infrastructure as Code Programs

Sep. 2024 • *Patrick Tser Jern Kon, Jiachen Liu, Yiming Qiu, Weijun Fan, Ting He Lei Lin, Haoran Zhang, Owen M. Park, George S. Elengikal, Yuxin Kang Ang Chen, Mosharaf Chowdhury, Myungjin Lee (Cisco), Xinyu Wang*

NeurIPS 2024
Rank A ICORE*

> “.....Notre évaluation révèle que **les modèles actuels, y compris GPT-4, obtiennent de faibles performances sur IaC-Eval,.....Cela souligne le besoin de progrès dans la génération de code IaC basée sur les LLM.** Nous rendons IaC-Eval open-source afin de faciliter les recherches futures dans ce domaine.”
>
> *Conclusion du papier de recherche*

Juil. 2021

Evaluating Large Language Models Trained on Code

OpenAI *Evaluating Large Language Models Trained on Code*

Juil. 2021 • ***Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, ...***

Aperçu : Ce document présente Codex (aujourd'hui GitHub Copilot), un modèle de langage dérivé de GPT-3, spécifiquement entraîné sur du code open-source issu de GitHub. L'objectif est d'évaluer sa capacité à générer du code fonctionnel à partir de descriptions textuelles.

LLM testés :

- **Codex** (GPT-3 mais entraîné spécifiquement sur du code provenant de GitHub)
- **Codex-S** (Codex Fine-tuning)
- **Et GPT-3, GPT-Neo, GPT-J...**

LLM évalués sur : Programmation générale

Pass@k : Mesure la capacité du modèle à résoudre un problème en générant jusqu'à k solutions, si au moins une des solutions passe les tests unitaires, le problème est considéré comme résolu.

Scénarios : 164 problèmes de programmation conçus manuellement (HumanEval)

	PASS@k		
	$k=1$	$k=10$	$k=100$
GPT-Neo 125M	0.75%	1.88%	2.97%
GPT-Neo 1.3B	4.79%	7.47%	16.30%
GPT-Neo 2.7B	6.41%	11.27%	21.37%
GPT-J 6B	11.62%	15.74%	27.74%
TabNine	2.58%	4.35%	7.59%
Codex-12M	2.00%	3.62%	8.58%
Codex-25M	3.21%	7.1%	12.89%
Codex-42M	5.06%	8.8%	15.55%
Codex-85M	8.22%	12.81%	22.4%
Codex-300M	13.17%	20.37%	36.27%
Codex-679M	16.22%	25.7%	40.95%
Codex-2.5B	21.36%	35.42%	59.5%
Codex-12B	28.81%	46.81%	72.31%

Alibaba Cloud

Nov. 2023

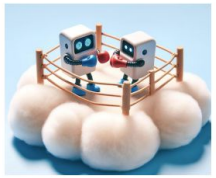
CloudEval-YAML: A Realistic and Scalable Benchmark

CloudEval-YAML: A Realistic and Scalable Benchmark

NeurIPS 2023 Workshop
Rank A* ICORE

Nov. 2023 • Yifei Xu (Alibaba Cloud et UCLA), Yuning Chen (Alibaba Cloud et UC Merced), Xumiao Zhang (University of Michigan), Xianshang Lin, Pan Hu, Yunfei Ma, Songwu Lu (UCLA), Wan Du (UC Merced), Z. Morley Mao (University of Michigan), Dennis Cai, Ennan Zhai

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

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. / Max
	pod	daemonset	service	job	deployment	others			
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

CloudEval-YAML: A Realistic and Scalable Benchmark

NeurIPS 2023 Workshop
Rank A* ICORE

Nov. 2023 • Yifei Xu (Alibaba Cloud et UCLA), Yuning Chen (Alibaba Cloud et UC Merced), Xumiao Zhang (University of Michigan), Xianshang Lin, Pan Hu, Yunfei Ma, Songwu Lu (UCLA), Wan Du (UC Merced), Z. Morley Mao (University of Michigan), Dennis Cai, Ennan Zhai

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:

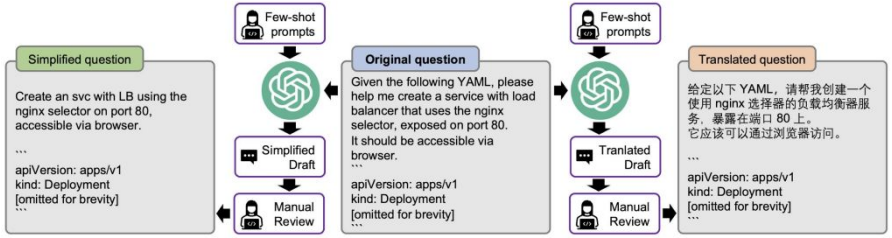
- **Simplified Question:** Short and clear language with domain-specific abbreviations
- **Translated Question:** Daily language used by Chinese cloud operation teams

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



CloudEval-YAML: A Realistic and Scalable Benchmark

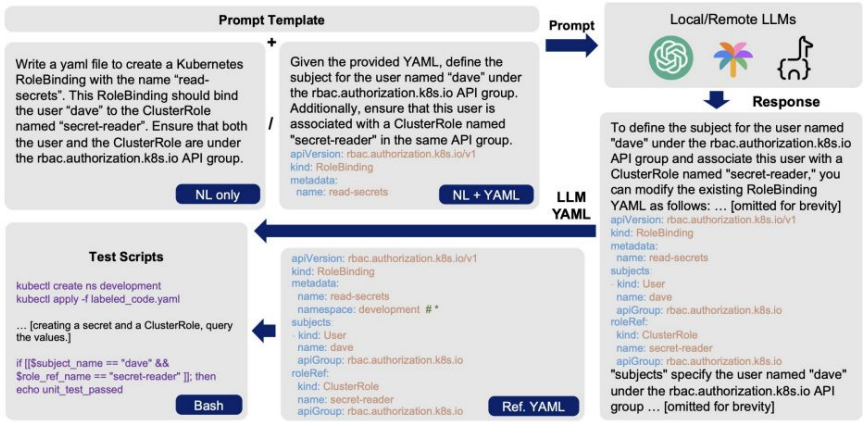
NeurIPS 2023 Workshop
Rank A* ICORE

Nov. 2023 • Yifei Xu (Alibaba Cloud et UCLA), Yuning Chen (Alibaba Cloud et UC Merced), Xumiao Zhang (University of Michigan), Xianshang Lin, Pan Hu, Yunfei Ma, Songwu Lu (UCLA), Wan Du (UC Merced), Z. Morley Mao (University of Michigan), Dennis Cai, Ennan Zhai

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



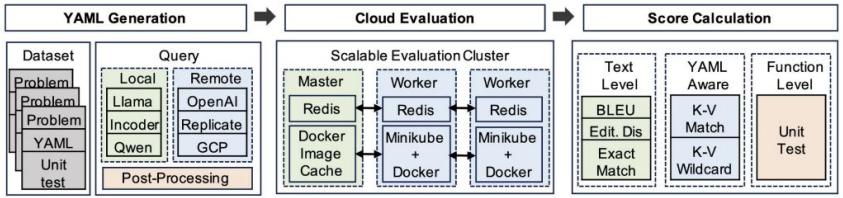
CloudEval-YAML: A Realistic and Scalable Benchmark

NeurIPS 2023 Workshop
Rank A* ICORE

Nov. 2023 • Yifei Xu (Alibaba Cloud et UCLA), Yuning Chen (Alibaba Cloud et UC Merced), Xumiao Zhang (University of Michigan), Xianshang Lin, Pan Hu, Yunfei Ma, Songwu Lu (UCLA), Wan Du (UC Merced), Z. Morley Mao (University of Michigan), Dennis Cai, Ennan Zhai

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

CloudEval-YAML: A Realistic and Scalable Benchmark

NeurIPS 2023 Workshop
Rank A* ICORE

Nov. 2023 • Yifei Xu (Alibaba Cloud et UCLA), Yuning Chen (Alibaba Cloud et UC Merced), Xumiao Zhang (University of Michigan), Xianshang Lin, Pan Hu, Yunfei Ma, Songwu Lu (UCLA), Wan Du (UC Merced), Z. Morley Mao (University of Michigan), Dennis Cai, Ennan Zhai

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

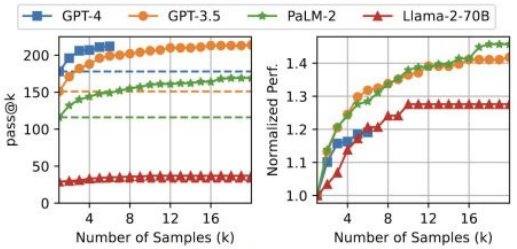
CloudEval-YAML: A Realistic and Scalable Benchmark

NeurIPS 2023 Workshop
Rank A* ICORE

Nov. 2023 • Yifei Xu (Alibaba Cloud et UCLA), Yuning Chen (Alibaba Cloud et UC Merced), Xumiao Zhang (University of Michigan), Xianshang Lin, Pan Hu, Yunfei Ma, Songwu Lu (UCLA), Wan Du (UC Merced), Z. Morley Mao (University of Michigan), Dennis Cai, Ennan Zhai

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)

Juin 2022

Generating Terraform Configuration Files with Large Language Models

Generating Terraform Configuration Files with Large Language Models

Juin 2022 • ***Oskar Bonde, Giacomo Verardo, Börje Ohlman (Ericsson)***

Aperçu : **Le projet explore la génération de fichiers de configuration Terraform en utilisant les LLM. L'étude inclut l'utilisation de deux stratégies (few-shot et fine-tuning).**

LLM testés : **GPT-2** (110M, 1.5B), **Codex** (12B), **CodeParrot** (110M, 1.5B)

LLM évalués sur : **Terraform**

Scénarios : **175 scénarios IaC AWS, GCP, et Azure** :

- **99 Functional Correctness** (Le **plan** Terraform généré doit être identique à celui de référence)
- **76 Compile Check** (Compile et produit le bon type de ressource)

	Functional Correctness				Compile Check			
	AWS	GCP	Azure	All	AWS	GCP	Azure	All
Size	49	29	21	99	31	24	21	76

Generating Terraform Configuration Files with Large Language Models

Juin 2022 • ***Oskar Bonde, Giacomo Verardo, Börje Ohlman** (Ericsson)*

Stratégies d'amélioration :

- Stratégies des exemples (Few-shot) pour Codex
- Réentraînement des modèles (Fine-tuning) pour GPT-2 et CodeParrot

Fine-tuning *(Pour GPT-2 et CodeParrot)* :

Tous les fichiers du BigQuery GitHub se terminant par .tf ont été sélectionnés.
Après filtrage, le jeu de données d'entraînement comprenait 23 838 fichiers.

Providers	AWS	GCP	Azure	All
Number of files	8617	5304	1140	23 838

"Clearly, the fine-tune dataset was too small to train a model with 1.5 billion parameters."

Generating Terraform Configuration Files with Large Language Models

Juin 2022 • **Oskar Bonde, Giacomo Verardo, Börje Ohlman** *(Ericsson)*

Résultat

Models	Functional Correctness (%)				Compile Check (%)			
	AWS	GCP	Azure	All	AWS	GCP	Azure	All
Codex-12B	60.16	23.17	47.52	46.65	67.81	2.67	46.10	41.26
CodeParrot-110M	13.80	10.97	9.14	11.98	24.45	16.42	8.48	17.50
CodeParrot-1.5B	6.12	0	9.52	5.05	20.19	8.92	0	11.05
GPT-2-110M	4.12	0	9.52	4.06	2.52	7.92	9.52	6.16
GPT-2-1.5B	0	0	4.67	0.99	0	0.42	3.05	0.97

Malgré l'absence de fine-tuning
Codex est en tête

Les performances de Codex illustrent que l'échelle du modèle (12B) jouent un rôle clé dans les résultats.

Nov. 2023

Exploring the Application of LLM in Infrastructure as Code

Exploring the Application of LLM in Infrastructure as Code

Nov. 2023 • ***Thibault Chanus, Michael Aubertin (SCC France)***

Aperçu : Ce document explore l'application des LLM pour la génération automatique de fichiers YAML.

LLM testés : **LLaMA** (7B, 13B), **LLaMA-2** (13B), **CodeUp** (13B), **Alpaca** (13B) **LLM évalués sur :** **Ansible YAML**

Scénarios :

Tâches d'orchestration impliquent principalement la génération de fichiers YAML pour Ansible, avec des configurations spécifiques, comme :

- Remontage des systèmes de fichiers en mode lecture seule
- Rechargement de la configuration sysctl
- Redémarrage de services critiques tels que auditd, fail2ban, sshd, et firewalld
- Désinstallation de packages non sécurisés

Exploring the Application of LLM in Infrastructure as Code

Nov. 2023 • ***Thibault Chanus, Michael Aubertin*** *(SCC France)*

Résultat :

“CodeUp se distingue par sa capacité à produire des fichiers YAML de qualité presque optimale. Toutefois, il convient de noter que, malgré ses performances impressionnantes, le modèle présente certaines lacunes, en particulier dans l’utilisation des modules spécifiques.”

“LLaMA-2 est un modèle exceptionnel qui a démontré son potentiel dans nos nombreux tests. Toutefois, dans notre contexte de génération de fichiers YAML pour Ansible, bien que les résultats soient prometteurs, ils ne sont pas encore entièrement satisfaisants.”

“Bien que LLaMA-13B représente une avancée significative par rapport à son homologue LLaMA-7B en termes de capacités et de précision, il est important de noter que la génération de fichiers YAML reste un défi. Malgré les progrès impressionnants réalisés par LLaMA-13B, nos expérimentations montrent que les fichiers YAML générés ne répondent toujours pas entièrement à nos exigences de qualité.”

02

Ajustement

Évolution des objectifs du projet à la lumière des travaux existants

LES ETAPES DU PROJET DE RECHERCE

~~03?~~

~~Benchmarking~~

~~Développement d'un outil d'évaluation des capacités des LLM~~

03

Benchmarking

~~Développement~~ **Enrichissement** d'un outil d'évaluation des capacités des LLM

~~04?~~

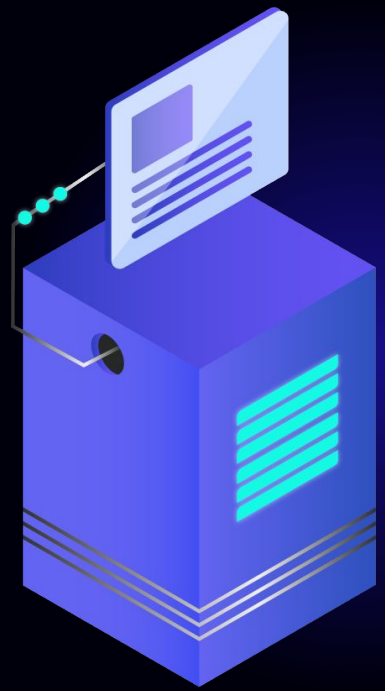
~~Stratégies~~

~~Expérimentation de stratégies pour améliorer les résultats~~

04

Stratégies

Expérimentation de **(nouvelles)** stratégies pour améliorer les résultats



Nouveaux objectifs

Réutiliser le benchmark open source* de l'étude IaC-Eval :

- **L'enrichir avec les mêmes scénarios, mais en français.**
  *Inspiration tirée de l'étude **CloudEval-YAML***
- **L'enrichir avec les mêmes scénarios, mais en version simplifiée.** *Inspiration tirée de l'étude **CloudEval-YAML***
- **L'enrichir avec de nouveaux LLM.**
- **Essayer de nouvelles stratégies pour améliorer les résultats.**
  Inspiration possible d'une étude du SAMOVAR
- **Faire une page web vitrine de notre projet afin de visualiser nos résultats.** *(Graph filtrables... ect)*

*MIT License

Scénarios cloud

DEFI N°1

Afin de tester les capacités des LLM à générer des fichiers de configuration pour le Cloud, nous avons besoin d'un vaste ensemble de scénarios (problématiques), ainsi que d'un mécanisme permettant d'évaluer et de valider la justesse des configurations produites.

Limiter les coûts

DEFI N°2

Le projet combine l'usage de ressources cloud et de LLM. Le défi des coûts est double, car il implique deux types de dépenses importantes :

- ~~Les coûts liés aux ressources cloud~~
- Les coûts associés à l'utilisation des API des LLM (hors LLM local)

UNIVERSITY OF MICHIGAN

Sep. 2024

IaC-Eval: A Code Generation Benchmark for Infrastructure as Code Programs

AVANT DE DÉMARRER

Avant de démarrer sur ses objectifs il faut tester le benchmark IaC-Eval: A Code Generation Benchmark for Infrastructure as Code Programs de l'Université du Michigan afin de s'assurer qu'il n'y a pas de points bloquants concernant son utilisation.

03

Benchmarking

Enrichissement d'un outil d'évaluation des capacités des LLM

Benchmarking

À venir dans les prochains mois

04

Stratégies

Expérimentation de nouvelles stratégies pour améliorer les résultats

Stratégies

À venir dans les prochains mois

MERCI !

Avez-vous des questions ?

pierre.laurent@telecom-sudparis.eu

adam.ouzegdouh@telecom-sudparis.eu