南京郵電大學
Nanjing University of Posts and Telecommunications

上下文学习

2025/06/11　　金鹏远

目录

背景 01

方法 02

讨论 03

总结 04

南京郵電大學 Nanjing University of Posts and Telecommunications

2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)

What Makes Good In-context Demonstrations for Code Intelligence Tasks with LLMs?

Shuzheng Gao[1†], Xin-Cheng Wen[1], Cuiyun Gao[1*], Wenxuan Wang[2], Hongyu Zhang[3], Michael R. Lyu[2]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

[2] Department of Computer Science and Engineering, The Chinese University of Hong Kong, China

[3] School of Big Data and Software Engineering, Chongqing University, China

szgao98@gmail.com, xiamenwxc@foxmail.com, gaocuiyun@hit.edu.cn, hyzhang@cqu.edu.cn, {wxwang,lyu}@cse.cuhk.edu.hk

背景

最近，人们越来越关注代码智能研究，旨在减轻软件开发人员的负担并提高编程生产力。随着大规模开源代码语料库和深度学习技术的进步，LLM在代码摘要、错误修复和程序合成在内的各种代码智能任务上取得了最先进的性能。

LLM展示的各种涌现能力中的一种是上下文学习 （ICL），它允许模型从特定上下文中的几个例子中学习，ICL在智能代码任务中也取得了极大的成效，但是目前缺乏对代码智能任务的 ICL 的深入研究。论文系统地分析了不同的演示构建方法如何影响 ICL 在代码智能任务上的性能。

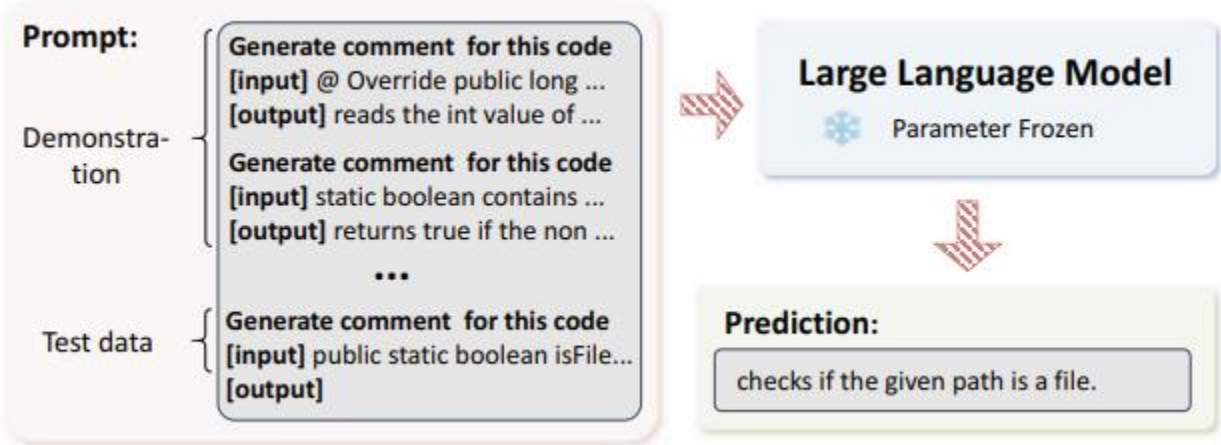
背景

RQ1：什么样的选择方法对代码智能任务中的 ICL 有帮助?

RQ2：在代码智能任务中，应该如何为 ICL 安排演示示例?

RQ3：提示中的演示示例数量如何影响代码智能任务中 ICL 的性能?

RQ4：研究结果的普遍性如何?

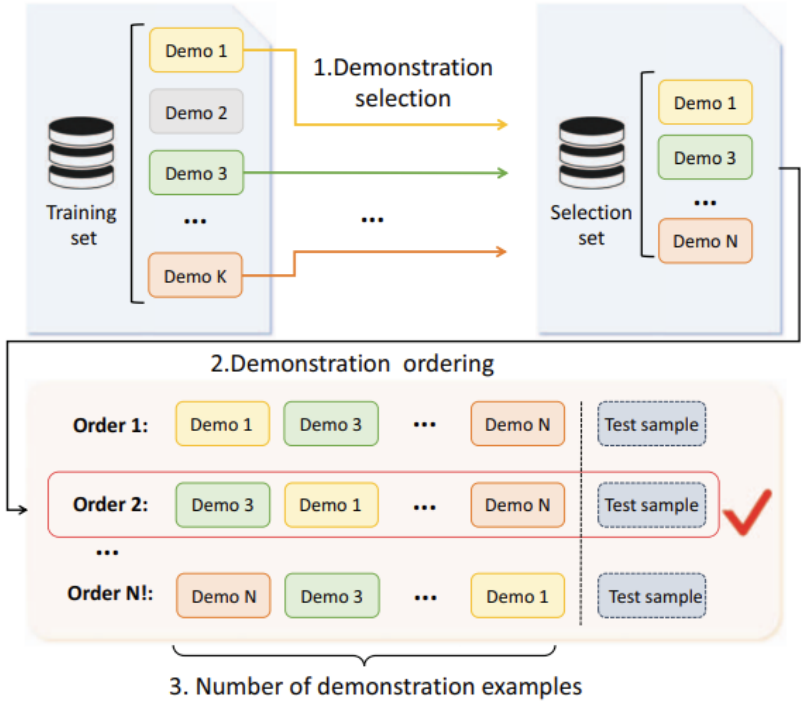


背景

优化大型预训练模型可能成本高昂且不切实际，尤其是当某些任务可用的微调数据有限时。ICL 提供了一种新的替代方案，它使用语言模型来执行下游任务，而无需更新参数。

任务级演示：对所有测试样本使用相同的演示示例，不考虑每个测试样本的差异。

实例级演示：针对不同的测试样本选择不同的演示示例。



方法

智能代码任务：

1）代码注释（Java）（BLEU-4，ROUGE-L，METEOR）

2）BUG修复（EM，BLEU-4）

3）程序合成：根据给定的自然语言描述自动生成源代码的任务。（Python）（Exact Match (EM)，CodeBLEU (CB)，Syntax Match (SM)，Dataflow Match (DM)）

模型：CodeX，GPT

Task	Datasets	Train	Dev	Test
Code Summarization	CSN-Java	164,923	5,183	10,955
	TLC	69,708	8,714	6,489
Bug Fixing	$B2F_{small}$	46,628	5,828	5,831
	$B2F_{medium}$	53,324	6,542	6,538
Program Synthesis	CoNaLa	2,389	-	500

Task	Template
Code Summarization	Generate comment (summarization) for this code [input] {#code} [output] {#comment}
Bug Fixing	Fix the bug according to the guidance [input] {#buggy code} <s> {#instruction} [output] {#fixed code}
Program Synthesis	Generate code based on the requirement [input] {#requirement}[output] {#code}

方法(RQ1)

任务级演示：

Random：从训练集中随机选择三组演示示例，并评估它们在不同任务上的表现。

KmeansRND：首先将整个样本分成 N 个集群，然后从每个集群中随机选择一个样本。

为了避免演示样本顺序对实验结果的影响，研究者对每个实验进行了三次不同顺序的测试，并报告每个指标的平均结果。此外，还通过变异系数（CV）来评估每种方法对不同顺序的敏感性。变异系数（CV）通过公式 σ/μ 计算，其中 σ 是标准差，μ 是平均值。CV 越低，表示数据的波动越小。

TABLE III: Experimental results of different demonstration selection methods on Code Summarization. "Avg" and "CV" denote the average results and Coefficient of Variation over three different orders, respectively.

Approach	Code Summarization											
	CSN						TLC					
	BLEU-4		ROUGE-L		METEOR		BLEU-4		ROUGE-L		METEOR	
	Avg	CV	Avg	CV	Avg	CV	Avg	CV	Avg	CV	Avg	CV
	Task-level Demonstration											
Random	19.64	1.44	35.46	1.88	15.30	1.54	17.29	0.71	34.28	0.61	12.48	0.67
KmeansRND	**20.71**	0.82	**38.03**	0.44	**16.34**	0.83	**17.91**	1.19	**35.69**	1.60	**13.48**	0.91
	Instance-level Demonstration											
BM-25	**22.35**	0.46	38.31	0.56	**17.01**	0.78	**36.96**	0.84	**51.42**	0.79	**24.22**	0.99
SBERT	22.27	0.23	**38.39**	0.42	16.91	0.22	36.42	0.61	50.47	0.40	23.86	0.68
UniXcoder	22.11	0.61	38.23	0.53	16.81	0.23	36.77	0.52	51.11	0.29	24.08	0.79
CoCoSoDa	21.92	0.46	37.85	0.22	16.78	0.24	36.91	0.69	50.69	0.53	24.08	0.39
Oracle (BM-25)	27.69	0.43	46.17	0.14	20.26	0.22	43.16	0.15	59.17	0.09	28.09	0.16

方法(RQ1)

Oracle：比较测试样本的输出和训练集中所有样本的输出之间的相似度，来找到与测试样本输出最接近的训练样本。这种方法可以确保所选的演示示例与测试任务具有高度的相关性。因为实际应用中通常无法事先知道测试样本的正确输出，Oracle 方法被视为性能的上限（理想情况）。

TABLE IV: Experimental results of different demonstration selection methods on Bug Fixing.

Approach	Bug Fixing							
	B2F$_{medium}$				B2F$_{small}$			
	BLEU-4		EM		BLEU-4		EM	
	Avg	CV	Avg	CV	Avg	CV	Avg	CV
	Task-level Demonstration							
Random	**86.96**	0.16	7.26	16.18	71.18	0.56	9.95	6.33
KmeansRND	86.91	0.17	**9.03**	5.45	**72.89**	1.36	**10.37**	3.86
	Instance-level Demonstration							
BM-25	**88.05**	0.09	**21.85**	1.78	**77.54**	0.13	**30.45**	0.96
SBERT	87.98	0.06	19.00	2.88	76.26	0.16	26.15	0.87
UniXcoder	87.87	0.09	19.14	2.00	77.52	0.07	29.93	0.51
CoCoSoDa	87.73	0.07	19.23	0.74	76.45	0.07	27.40	1.04

TABLE V: Experimental results of different demonstration selection methods on Program Synthesis.

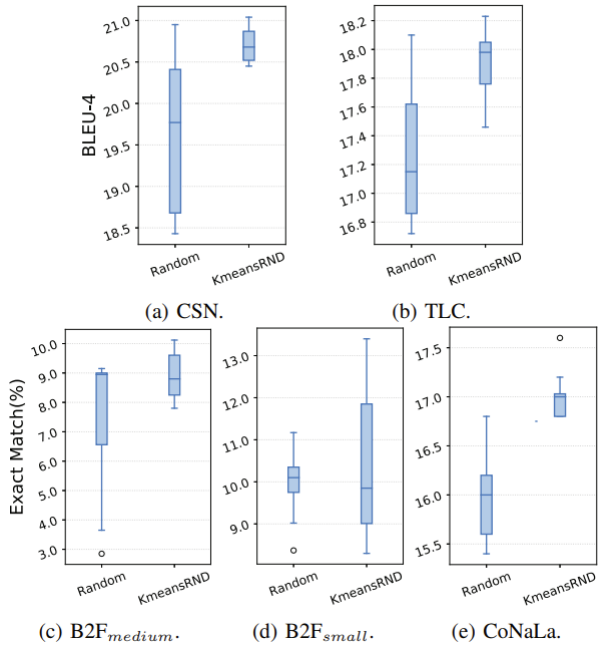
Approach	Program Synthesis							
	CB		SM		DM		EM	
	Avg	CV	Avg	CV	Avg	CV	Avg	CV
	Task-level Demonstration							
Random	**28.36**	1.30	44.37	0.83	**39.70**	1.33	16.00	1.60
KmeansRND	28.03	1.47	**44.41**	0.54	37.31	1.54	**17.03**	1.06
	Instance-level Demonstration							
BM-25	**30.37**	0.91	**46.22**	0.84	40.75	1.06	**18.53**	0.50
SBERT	29.08	0.70	44.91	0.31	39.81	3.01	16.13	2.54
UniXcoder	28.96	0.50	43.93	0.67	37.96	1.12	16.00	3.53
CoCoSoDa	29.42	0.82	44.62	0.70	**40.91**	1.12	16.30	0.86

方法(RQ1)

1.示例的多样性有助于 ICL 的示范选择。它可以帮助提高整体性能，并针对不同的示例组进行更稳定的预测。

2.示例的检索方法会影响 ICL 的性能，其中BM-25是一种简单有效的方法。

3.与任务级演示相比，实例级演示可以取得更好的性能，并且通常对示例顺序的变化更健壮。



(a) CSN. (b) TLC.

(c) B2F$_{medium}$. (d) B2F$_{small}$. (e) CoNaLa.

方法(RQ2)

任务级演示选择:Random和KmeansRND

实例级演示选择:BM25

Approach		Code Summarization (CSN)			Bug Fix ($B2F_{small}$)		Program Synthesis (CoNaLa)			
		BLEU-4	ROUGE-L	METEOR	BLEU-4	EM	CB	SM	DM	EM
Random	Random	20.46	36.71	16.17	**72.40**	9.52	27.72	44.46	37.53	15.53
	Similarity	**21.04**	**37.86**	**16.26**	72.02	**9.93**	**28.47**	**44.87**	37.79	**16.00**
	Reverse Similarity	19.78	33.71	15.64	71.44	9.02	27.62	44.48	**37.96**	15.20
KmeansRND	Random	20.67	**37.64**	15.97	72.29	8.60	26.64	42.97	**37.24**	**16.87**
	Similarity	**20.69**	37.62	16.05	**72.90**	**10.15**	**27.20**	42.97	36.93	16.40
	Reverse Similarity	20.55	37.43	**16.20**	72.05	9.78	27.09	**43.74**	37.19	16.60
BM-25	Random	**22.35**	**38.31**	**17.01**	77.54	30.45	30.37	46.22	40.75	**18.53**
	Similarity	22.23	38.12	**17.01**	**77.76**	**30.95**	**30.83**	**46.41**	**41.33**	17.60
	Reverse Similarity	22.13	38.26	16.91	77.60	29.80	30.01	45.72	39.60	18.20

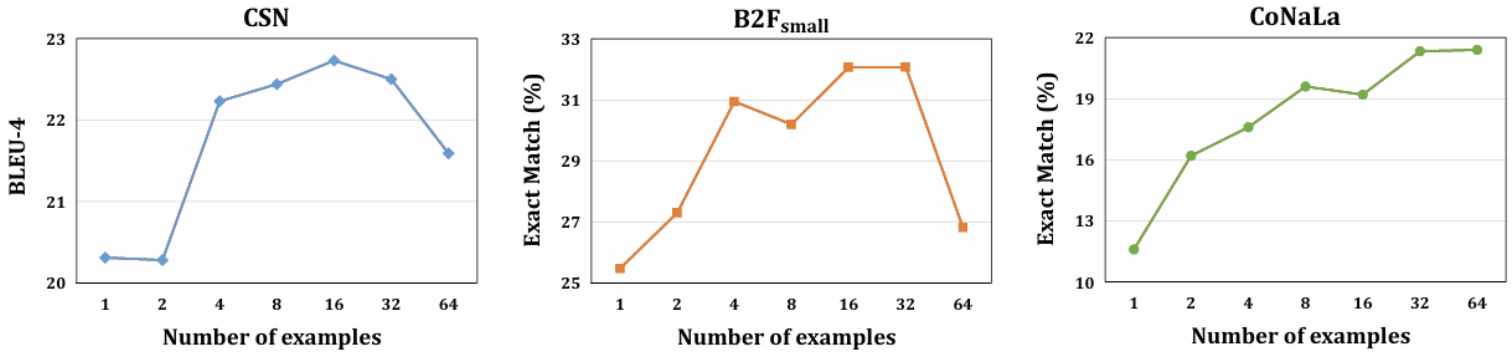
发现:

演示示例的不同顺序会影响 ICL 的性能。在大多数情况下，演示示例测试示例的相似度按升序排列可以获得相对较好的结果。

在某些情况下，Similarity 和 Reverse Similarity 的表现都比使用随机顺序的平均结果差，这表明更复杂的演示排序方法可以在未来的工作中探索。

方法(RQ3)

考虑到截断问题，提示中的更多演示示例并不总是能带来更好的性能。为了节省成本，建议在演示中使用四个示例。



(a) Code Summarization. (b) Bug Fixing. (c) Program Synthesis.

Fig. 4: Experimental results of ICL with different number of demonstration examples.

总结

演示的多样性、相关性、排序方式、数量等对于LLM做出相应判断均有影响，可进一步考虑研究动态排序算法，提高演示对于LLM的指导作用。

敬请批评指正