南京大学 NANJING UNIVERSITY

奋进行动 团结奋斗 争先进位 坐言起行 应势而动

科研实践期末报告

吴林洋

CONTENTS 目录

01 Binary Code Embedding

应用

恶意软件检测
代码相似性检测
逆向工程
漏洞检测
软件分类

Transformed-based优点

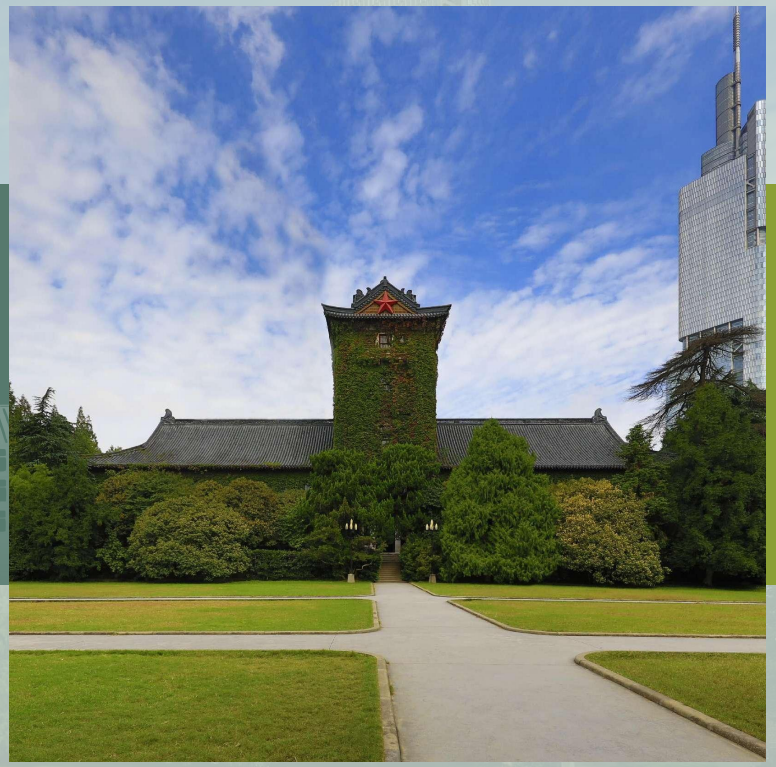
和手工制作的embedding相比：
代价低，适用于大量代码
可移植性强

和基于学习的embedding相比：
CNN、RNN:较难建模长距离的依赖关系

问题

目前的一些模型仅仅使用自然语言的处理方法
缺乏对指令边界的理解
缺乏对指令跳转的理解
缺少数据集

02 精读论文



J-Trans

Jump-Awre Transformer for
Binary Code Similarity

K-Trans

Knowledge-Awre Transformer
for Binary Code Embedding

Jump-Awre Transformer for Binary Code Similarity

预处理数据

预训练

微调

模型评估

预处理数据

解决问题：
1.Transfor只能处理固定词汇表上的数据
2.二进制代码缺少跳转指令

处理方法
1.将字符串字面量替换为特殊标记 <str>
2.将常量值替换为特殊标记 <const>
3.保留外部函数调用的名称和标签作为标记，并将内部函数调用的名称替换为 <function>
4.对于每个跳转对，将源标记替换为 JUMP_XXX，其中 XXX 是跳转目标的顺序编号

预处理示例

0x1000: mov eax, 100
0x1005: call printf
0x100A: jmp 0x2000
0x2000: mov ebx, 200
0x2005: call internal_func
0x200A: jmp 0x1000

1：mov eax, <const>
2：call printf
3：jmp JUMP_4
4：mov ebx, <const>
5：call <function>
6：jmp JUMP_1

非监督预训练

Masked Language Model
mov eax, <const>--->[MASK] eax, <const>

Jump Target Prediction
jump JUMP_1---> <LOC>

Knowledge-Aware Transformer for Binary Code Embedding

Method	ISA Knowledge	Instruction Boundary	Implicit Dependency	Contextual Inference
word2vec	N	N	N	N
PalmTree	N	N	Partial	N
BinBert	N	N	N	Y
jTrans	N	Partial	Partial	Y
kTrans	Y	Y	Y	Y

预处理

固定大小单词表处理

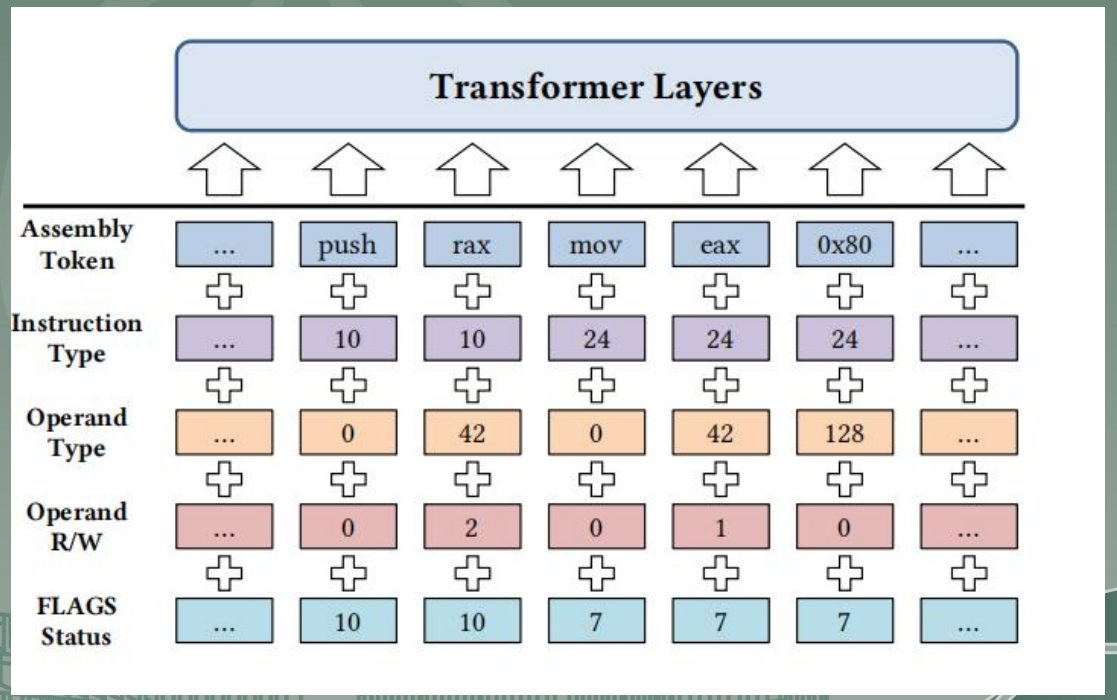
显式knowledge注入：

（1）指令类型

（2）操作数类型

（3）操作数读写状态

（4）状态寄存器



隐式知识注入

token-level Mask

mov eax, 1 | add eax, ebx | jmp 0x1000 | call printf
mov, eax, 1, [MASK], eax, ebx, jmp, [MASK], call, printf

instruction-level Mask

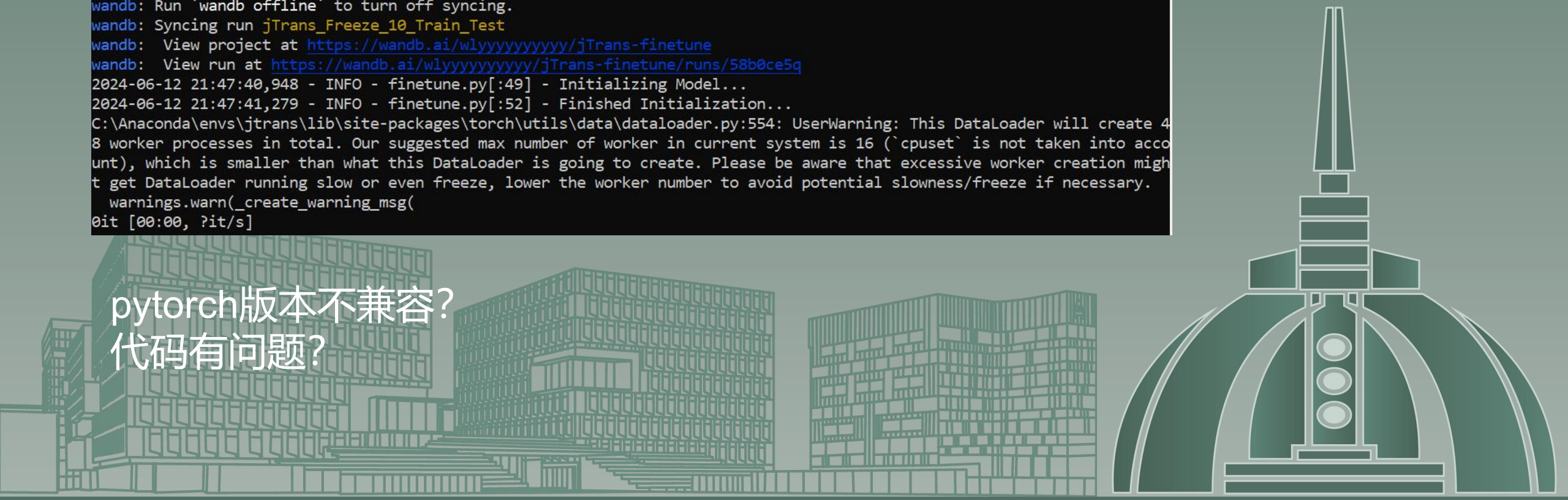
mov eax, 1 | add eax, ebx | jmp 0x1000 | call printf
mov eax, 1 | [MASK] | jmp 0x1000 | call printf

03 代码复现

jTrans有开源代码
主要是配置环境

```
2024-06-12 21:47:31,477 - INFO - finetune.py[:236] - Done ...
2024-06-12 21:47:31,487 - INFO - finetune.py[:238] - Tokenizer Done ...
TOTAL  0
TOTAL  0
2024-06-12 21:47:31,562 - INFO - finetune.py[:248] - Done ...
wandb: Currently logged in as: 1801632157 (wlyyyyyyyyyy). Use `wandb login --relogin` to force relogin
wandb: Tracking run with wandb version 0.17.1
wandb: Run data is saved locally in C:\Users\yaoyang12326\jTrans\wandb\run-20240612_214734-58b0ce5q
wandb: Run `wandb offline` to turn off syncing.
wandb: Syncing run jTrans_Freeze_10_Train_Test
wandb:  View project at https://wandb.ai/wlyyyyyyyyyy/jTrans-finetune
wandb:  View run at https://wandb.ai/wlyyyyyyyyyy/jTrans-finetune/runs/58b0ce5q
2024-06-12 21:47:40,948 - INFO - finetune.py[:49] - Initializing Model...
2024-06-12 21:47:41,279 - INFO - finetune.py[:52] - Finished Initialization...
C:\Anaconda\envs\jtrans\lib\site-packages\torch\utils\data\dataloader.py:554: UserWarning: This DataLoader will create 4
8 worker processes in total. Our suggested max number of worker in current system is 16 (`cpuset` is not taken into acco
unt), which is smaller than what this DataLoader is going to create. Please be aware that excessive worker creation migh
t get DataLoader running slow or even freeze, lower the worker number to avoid potential slowness/freeze if necessary.
  warnings.warn(_create_warning_msg(
0it [00:00, ?it/s]
```

pytorch版本不兼容?
代码有问题?



南京大学
NANJING UNIVERSITY

谢谢观看！

日期：6.13