[论文阅读] (27) AAAI20 Order Matters: 基于图神经网络的二进制代码相似性检测（腾讯科恩实验室）

《娜璋带你读论文》系列主要是督促自己阅读优秀论文及听取学术讲座，并分享给大家，希望您喜欢。由于作者的英文水平和学术能力不高，需要不断提升，所以还请大家批评指正，非常欢迎大家给我留言评论，学术路上期待与您前行，加油。

前一篇文章介绍Excel论文可视化分析基础知识。这篇文章将带来AAAI20腾讯科恩实验室的经典工作——Order Matters，提出语义感知（Semantic-Aware）神经网络来实现二进制代码相似性检测，希望这篇文章对您有所帮助。一方面自己英文太差，只能通过最土的办法慢慢提升，另一方面是自己的个人学习笔记，并分享出来希望大家批评和指正。这些大佬是真的值得我们去学习，献上小弟的膝盖~fighting!

- 问题：究竟怎么实现语义感知？又如何与二进制代码相结合？
- 感受：这篇文章和自己的写作及研究风格真心像，建议以后深入学习和复现。

The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)

Order Matters: Semantic-Aware Neural Networks for Binary Code Similarity Detection

Zeping Yu,[1*] Rui Cao,[1*] Qiyi Tang,[1] Sen Nie,[1] Junzhou Huang,[2] Shi Wu[1†]
[1]Tencent Security Keen Lab, Shanghai, China
[2]Tencent AI Lab, Shenzhen, China
{zepingyu, ruicho}@foxmail.com, {dodgetang, snie, joehhuang, shiwu}@tencent.com

原文作者：Zeping Yu, Rui Cao, Qiyi Tang, Sen Nie, Junzhou Huang, Shi Wu
原文标题：Order Matters: Semantic-Aware Neural Networks for Binary Code Similarity Detection
原文链接：https://ojs.aaai.org/index.php/AAAI/article/view/5466
发表会议：AAAI 2020
官方博客：科恩 - AAAI-20论文解读：基于图神经网络的二进制代码分析

除了原文和作者的理解，本文还参考了下面两位老师的博客，再次感谢，向老师和好友们学习。

- Order Matters: SANN二进制代码相似性检测
- Order Matters: Semantic-Aware Neural Networks for Binary Code Similarity Detection

> **作者感受：**
> **这篇论文的框架风格和我的很像，非常值得我学习，尤其是安全和AI的结合。同时，文中的英文表述、创新点、模型设计以及实验评估也值得学习。**

腾讯科恩实验室官方博客：

- https://keenlab.tencent.com/zh/index.html

- https://github.com/KeenSecurityLab/BinAbsInspector

文章目录

前文赏析:

- [论文阅读] (01) 拿什么来拯救我的拖延症？初学者如何提升编程兴趣及LATEX入门详解
- [论文阅读] (02) SP2019-Neural Cleanse: Identifying and Mitigating Backdoor Attacks in DNN
- [论文阅读] (03) 清华张超老师 - GreyOne: Discover Vulnerabilities with Data Flow Sensitive Fuzzing
- [论文阅读] (04) 人工智能真的安全吗？浙大团队外滩大会分享AI对抗样本技术
- [论文阅读] (05) NLP知识总结及NLP论文撰写之道——Pvop老师
- [论文阅读] (06) 万字详解什么是生成对抗网络GAN？经典论文及案例普及
- [论文阅读] (07) RAID2020 Cyber Threat Intelligence Modeling Based on Heterogeneous GCN
- [论文阅读] (08) NDSS2020 UNICORN: Runtime Provenance-Based Detector for Advanced Persistent Threats
- [论文阅读] (09)S&P2019 HOLMES Real-time APT Detection through Correlation of Suspicious Information Flow
- [论文阅读] (10)基于溯源图的APT攻击检测安全顶会总结

- [论文阅读] (11)ACE算法和暗通道先验图像去雾算法（Rizzi | 何恺明老师）
- [论文阅读] (12)英文论文引言introduction如何撰写及精句摘抄——以入侵检测系统(IDS)为例
- [论文阅读] (13)英文论文模型设计（Model Design）如何撰写及精句摘抄——以入侵检测系统(IDS)为例
- [论文阅读] (14)英文论文实验评估（Evaluation）如何撰写及精句摘抄（上）——以入侵检测系统(IDS)为例
- [论文阅读] (15)英文SCI论文审稿意见及应对策略学习笔记总结
- [论文阅读] (16)Powershell恶意代码检测论文总结及抽象语法树（AST）提取
- [论文阅读] (17)CCS2019 针对PowerShell脚本的轻量级去混淆和语义感知攻击检测
- [论文阅读] (18)英文论文Model Design和Overview如何撰写及精句摘抄——以系统AI安全顶会为例
- [论文阅读] (19)英文论文Evaluation（实验数据集、指标和环境）如何描述及精句摘抄——以系统AI安全顶会为例
- [论文阅读] (20)USENIXSec21 DeepReflect：通过二进制重构发现恶意功能（恶意代码ROI分析经典）
- [论文阅读] (21)S&P21 Survivalism: Systematic Analysis of Windows Malware Living-Off-The-Land (经典离地攻击)
- [论文阅读] (22)图神经网络及认知推理总结和普及-清华唐杰老师
- [论文阅读] (23)恶意代码作者溯源(去匿名化)经典论文阅读：二进制和源代码对比
- [论文阅读] (24)向量表征：从Word2vec和Doc2vec到Deepwalk和Graph2vec，再到Asm2vec和Log2vec（一）
- [论文阅读] (25)向量表征经典之DeepWalk：从Word2vec到DeepWalk，再到Asm2vec和Log2vec（二）
- [论文阅读] (26) 基于Excel可视化分析的论文实验图表绘制总结——以电影市场为例
- [论文阅读] (27) AAAI20 Order Matters: 二进制代码相似性检测（腾讯科恩实验室）

一.摘要

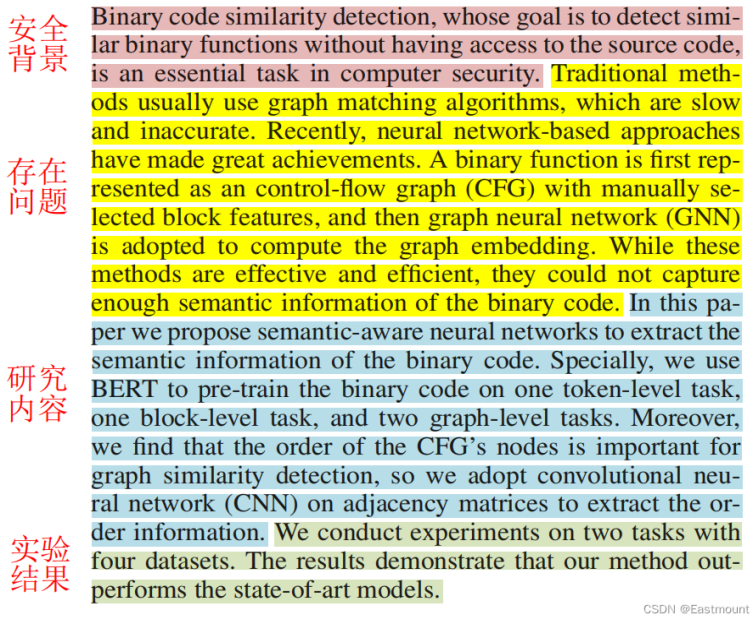
二进制代码相似性检测是计算机安全领域中的一项重要（essential）任务，其目标是检测二进制函数的相似性。

传统方法通常使用图匹配（graph matching）算法，但速度缓慢且不准确（inaccurate）。近年来，基于神经网络的研究方法已取得巨大的成就（have made great achievements）。首先将一个二进制函数表示为一个具有手动选择块特征的控制流图（CFG），然后采用图神经网络（GNN）来计算图嵌入（graph embedding）。虽然这些方法非常有效（effective and efficient），但它们不能足够地捕获二进制代码的语义信息。

本文提出一种语义感知（semantic-aware）神经网络来提取二进制代码的语义信息，其输入是以基本块为单位的CFG图。具体而言，我们使用BERT在一个令牌级任务（token-level）、一个块级任务（block-level）和两个图级任务（graph-level）上对二进制代码进行预训练。此外，我们发现CFG节点的顺序对于图的相似度检测很重要，因此我们在邻接矩阵上采用卷积神经网络（CNN）来提取顺序信息。

- 首先使用Bert生成每一个基本块的embedding，接着使用MPNN网络生成CFG图的整体embedding，同时融入通过邻接矩阵生成的CFG图顺序信息，经过MLP得到最终的embedding值，以此进行相似度比较。

我们用四个数据集在两个任务上进行了实验。结果表明，我们的方法优于最先进（state-of-art）的模型。



二.引言

> 由于每篇论文的引言都非常重要，会告诉大家为什么有这个工作，以及这个工作做了什么，有什么贡献。因此该部分作者会全文翻译，后续章节则介绍重点内容。

二进制代码相似性检测（Binary code similarity detection） 旨在检测两个给定的二进制函数是否相似 [不访问源代码]。二进制代码分析被广泛应用于计算机安全领域，譬如代码克隆检测（code clone detection）、漏洞发现（vulnerability discovery）、恶意软件检测等。

- 科恩："同一份源代码在不同编译器，不同平台，不同优化选项的条件下所得到的二进制代码是不相同的，我们的任务目标是把同一份源代码所编译出的不同的二进制代码找到。"

传统方法采用 **图匹配算法**（Liu et al. 2006）来计算两个函数的相似度。然而，这些基于图匹配的方法速度缓慢，并且可能很难适应不同的应用程序。随着近年来深度学习算法的发展，研究者尝试在控制流图（CFG）上使用图神经网络算法，并取得不错的效果。

文献[1]提出一种基于神经网络的方法 `Gemine` （Xu et al.2017），它的输入是两个二进制函数的pair，输出是这两个二进制函数的相似度得分。该工作证明 `Gemine` 的准确率和速度优于现有方法。

- 首先，将二进制函数的控制流图（CFG）作为输入，并使用人工设计的特征提取方法将每个block表示成低维的向量， `Gemini` 会将其转换为一个有属性的CFG。如图1所示。
- 其次，使用 `Structure2vec` 算法（Dai, Dai, and Song 2016）生成graph embedding。
- 最后，使用siamese网络计算相似度得分并使用梯度下降算法降低损失训练模型。

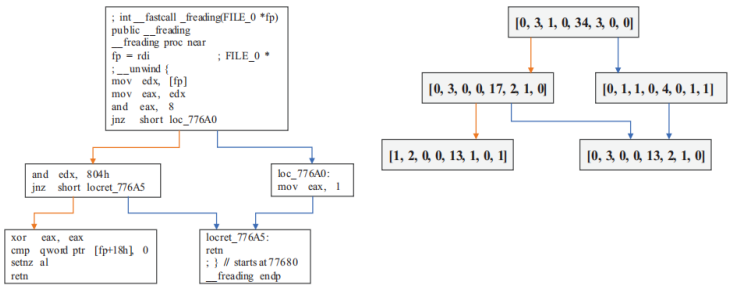


Figure 1: An example of a control flow graph (CFG) and its manually selected block features.

> [1] Xu X.; Liu C.; Feng Q.; et al. 2017. Neural network-based graph embedding for crossplatform binary code similarity detection. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), 363–376. ACM.

尽管基于神经网络的模型已经取得了很大的进步，但仍存在一些未被考虑的问题。

- 首先，如图1所示，每个block都被表示为一个低维向量，这个特征提取是人工设计的，在Gemini中block特征只有8维向量，这个压缩的过程会损失很多语义信息。
- 其次，节点的顺序在表示二进制函数中起着重要的作用，而以往的方法并没有设计特定的算法提取这一特征。

为了解决这两个问题，我们提出一个包含三个组件的整体框架：

- 语义感知模块（semantic-aware modeling）
- 结构感知模块（structural-aware modeling）
- 顺序感知模块（order-aware modeling）

> **原文（优美句子）**：
> Even though neural network-based models have achieved a lot, there are several important things that have not been taken into consideration. Firstly, as shown in Figure 1, each block is represented as a low-dimensional embedding with manually selected features, which will cause the loss of much semantic information. Secondly, the order of the nodes plays an important role in representing binary functions, while previous approaches did not design methods to extract it. To solve these two problems, we propose an overall framework with three components: semantic-aware modeling, structural-aware modeling, and order-aware modeling.

(1) 在语义感知模块：

我们使用NLP模型来提取二进制代码的语义信息。CFG块中的token被视为单词，CFG块被视为句子。 在先前的工作中：

- （Massarelli et al. 2019）使用word2vec模型训练块中的token embeddings，然后使用注意机制获得block embedding。
- （Zuo et al. 2018）借鉴了神经机器翻译（NMT）的思想来学习跨平台二进制码之间的语义关系。

在本文中，我们采用BERT（Devlin et al. 2018）对tokens和blocks进行预训练。与BERT相同，我们对MLM（masked language model）任务的标记进行预训练，并提取所有相邻块对邻接节点预测任务（ANP）进行预训练。与分别学习token向量和block向量不同，**本文方法能够同时学习token向量和block向量**。此外，因为我们的最终目标是生成完整的图表示，所以我们添加了两个图级任务。

- 一种是确定两个采样块是否在同一个图中，我们称之为图内块任务（BIG，block inside graph task）。
- 另一种是区分块属于哪个平台/优化选项，称为图分类任务（GC，graph classification task）。

我们发现，额外的任务可以帮助提取更多的语义信息，更好地学习块表示。在对块嵌入进行预训练之后，我们将在图级任务上对它们进行微调。

(2) 在结构感知模块：

我们使用MPNN（Gilmer等人2017）和GRU（Cho等人2014）更新函数。（Xu et al. 2018）已经证明了图神经网络可以具有像Weisfeiler-Lehman测试一样的区分能力。我们发现，在每个步骤中使用GRU比只使用tanh函数可以存储更多的信息。

(3) 在顺序感知模块：

我们尝试设计一种体系结构来提取CFG的节点顺序信息。图2显示函数"_freading"在不同平台x86-64和ARM上编译出的二进制代码的控制流图及邻接矩阵。这两个控制流图的节点顺序是非常相似的，例如node1都与node2和node3相连，node2都与node4和node5相连，而这种相似性可以体现在它们的邻接矩阵上。

通过探索了许多跨平台函数对，我们发现节点顺序的变化很小。在此基础上，我们提出了一种简单的捕获顺序信息的方法，即在邻接矩阵上使用CNN。**我们发现只有一个三层的CNN表现良好。** 我们进一步探索了其他CNN模型，如Resnet（He et al. 2016），并讨论了CNN模型可以从邻接矩阵中学到什么。

> O(∩_∩)O
> 哈哈，学习如何证明三层CNN更好，想到自己的论文。

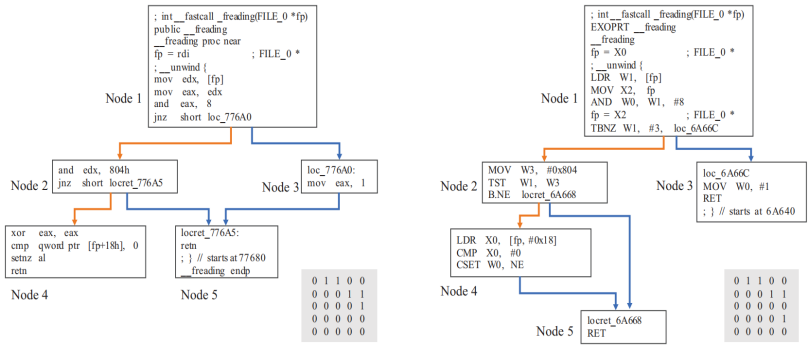


Figure 2: Two CFGs and their adjacency matrices of function "_freading" on different platforms (x86-64 & ARM).

本文的贡献如下：

- 我们提出了一个学习CFG图嵌入的通用框架，它可以学习语义信息、结构信息和顺序信息。
- 在语义感知模块中，我们采用BERT对MLM任务和相邻节点预测（ANP）任务进行token向量（token embeddings）和block向量（block embeddings）的预训练。此外，我们还添加了两个图级任务（graph-level tasks）来更好地学习块表示，分别是图块内部任务（BIG）和图分类任务（GC）。
- 在顺序感知模型中，我们发现节点顺序是有用的。我们采用了在邻接矩阵上的CNN模型来提取CFGs的节点顺序信息，并取得了很大的成绩。然后，我们探索CNN可以从邻接矩阵中学到什么。
- 我们在四个数据集的两个任务上进行实验，结果表明，我们提出的模型比以前的方法取得了更好的性能。

Our contributions are as follows:

1) We propose a general framework to learn graph embeddings of CFGs, which could learn semantic information, structural information, and order information.

2) In semantic-aware modeling, we adopt BERT to pre-train token embeddings and block embeddings with masked language model task (MLM) and adjacent node prediction task (ANP). Additionally we add two graph-level tasks to learn block representation better, which are block inside graph task (BIG) and graph classification task (GC).

3) In order-aware modeling, we find that the node order is useful. We adopt CNN models on the adjacency matrices to extract the node order information of CFGs, which makes great achievements. Then we explore what CNN could learn from adjacency matrices.

4) We conduct experiments on two tasks with four datasets, and the results demonstrate that our proposed model achieves much better performance than the previous methods.

三.相关工作

1.Graph Neural Networks

图神经网络提出来学习节点表示和图表示。典型方法包括：

- GCN：使用卷积层来更新节点嵌入
- GraphSAGE：采用聚合函数将节点与其相邻节点进行合并
- GAT：利用注意机制从重要节点接收到更多的信息

2.BERT

BERT是自然语言处理中最先进的预训练模型，通过Transformer实现。

- MLM
- NSP

3.Binary Code Similarity Detection

二进制代码相似度检测是计算机安全研究中的一项重要任务。传统的方法使用图匹配算法来计算图的相似度。然而，这些方法是缓慢和低效的。现有方法缺陷：

- 获取相似块对是一个有监督的过程，需要专家经验和领域知识，以及一些块不能唯一标注。
- 在实际使用中，需要针对不同的平台组合训练不同的模型。

四.本文方法

1.Overall Structure

本文模型的输入是二进制代码的控制流图，其中每个块都是一个带有中间表示的令牌序列。模型的总体结构如图3所示，包含semantic-aware 模块、structural-aware模块、order-aware模块。

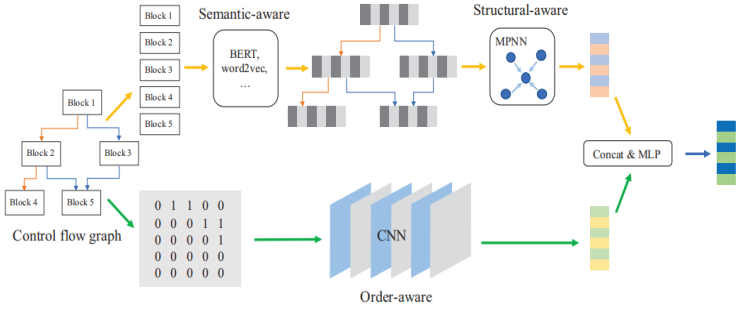


Figure 3: Overall structure of our model. The model has three components: semantic-aware modeling, structural-aware modeling and order-aware modeling.

- 在语义感知模块，该模型将CFG作为输入，并使用BERT对token embedding预训练，得到block embedding。
- 在结构感知模块，我们使用MPNN和GRU更新函数来计算图的语义和结构向量（graph semantic & structural embedding），记为 g_{ss}。
- 在顺序感知模块，模型以CFG的邻接矩阵为输入，采用CNN计算graph order embedding，记为 g_o。

最后，对两个向量使用concat和MLP得到最终的graph embedding，如公式1所示。

$$g_{final} = \text{MLP}([g_{ss}, g_o]) \tag{1}$$

2.Semantic-aware Modeling

在语义感知模块中，我们提出了一个包含4个任务的BERT预训练模型来处理CFG。这个模型有几个优点。

- 首先，可以从不同平台、不同架构、基于同一模型的不同编译优化选项生成的不同CFG中提取块向量。
- 其次，可以从训练前的过程中得到令牌级、块级和图级的信息，因为我们有一个令牌级任务、一个块级任务和两个图级任务。
- 最后，训练过程完全基于CFG图，不需要修改编译器或其他操作来获得相似的块对。

本文方法的灵感来自于NLP中的句子嵌入任务，CFG中的块可以看作句子，块中的token可以看做单词。 这个任务是提取一个句子的embedding，完成这个任务主要有两种方法。

- 监督方法，如文本分类训练（Joulin et al. 2016）。
- 无监督的方法，如n-gram特征和decoder-encoder skip思想（Kiros et al. 2015）。

我们使用基于BERT的改进模型来提取CFG上的块向量。如图4所示，在我们的训练前过程中有四个任务：

- Masked language model（MLM）
- Adjacency node prediction（ANP）
- Block inside graph（BIG）
- Graph classification（GC）

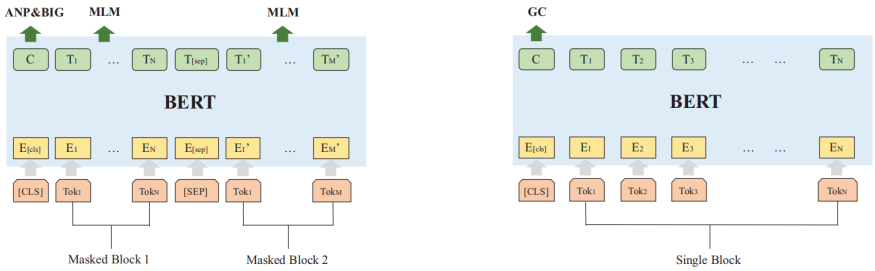


Figure 4: Bert with 4 tasks: MLM, ANP, BIG and GC.

对于节点内的token序列，我们使用MLM来提取块内的语义信息。MLM是一个令牌级任务，它屏蔽输入层上的令牌，并在输出层上预测它们，和语言模型的方式相同。

邻接节点预测任务（ANP）是一个块级的任务。在图中，块的信息不仅与块本身的信息有关，还与块本身的邻居信息有关，我们希望模型能够学习这些信息。在ANP任务中，在一个图上提取所有相邻的块，并在同一个图中随机抽取几个块，以预测两个块是否相邻。这两个任务（MLM & ANP）类似于原始BERT论文中的MLM & NSP任务（Devlin et al. 2018）。

为了更好地利用graph-level的信息，我们添加了两个辅助监督任务BIG和GC。

- BIG任务与ANP的方式类似，区别是pair的正负例选择方式不同。BIG任务试图让模型判断两个block是否在同一个图中，希望模型可以尽可能地学到此信息，从而对我们的graph-level任务有帮助。因此，在BIG任务中同图的block pair为正例，不同图的block pair为负例。
- GC为graph-level的block分类任务，在我们的场景中，在不同平台、不同编译器、不同优化选项的条件下，得到的block信息有所不同，我们希望模型可以让block embedding中包含这种信息。GC对block进行分类，判断block属于哪个平台，哪个编译器，以及哪个优化选项。

3.Structural-aware Modeling

在从BERT预训练中获得块向量后，我们使用MPNN来计算每个CFG的graph semantic & structural embedding。MPNN有三个步骤：message function（M），update function（U）以及readout function（R）。具体步骤如公式2-公式4所示。

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}) \tag{2}$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \tag{3}$$

$$g_{ss} = R(h_v^T | v \in G) \tag{4}$$

其中，G表示整个图，v表示节点，N(v)表示v的相邻节点。在本文的场景中，节点即是控制流图中的block，图即是经过预训练后表示成block向量的控制流图。本文在message步骤使用MLP，update步骤使用GRU，readout步骤使用sum，如公式5-公式7所示。

$$m_v^{t+1} = \sum_{w \in N(v)} \mathrm{MLP}(h_w^t) \tag{5}$$

$$h_v^{t+1} = \mathrm{GRU}(h_v^t, m_v^{t+1}) \tag{6}$$

$$g_{ss} = \sum_{v \in G} \mathrm{MLP}(h_v^0, h_v^T) \tag{7}$$

4.Order-aware Modeling

在这个模块中，我们的目标是提取CFG节点的顺序信息，本文使用CNN模型来观察能学到哪些信息。图5显示三个图（块中没有语义信

息）和它们的邻接矩阵，它们可以通过添加几个小变化来相互传递。这三个图非常相似，每个图中都有一个三角形特征（图a的节点123，图b的节点234，图c的节点134），这个特征体现在它们的邻接矩阵中。

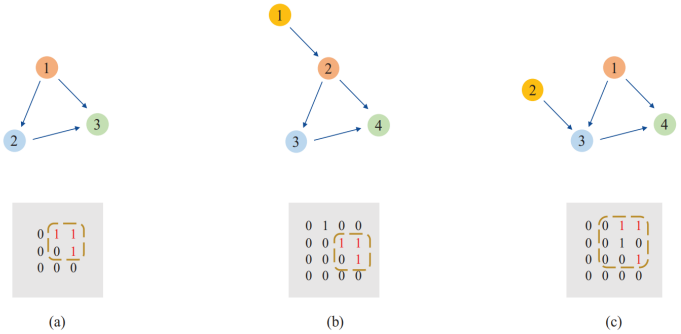


Figure 5: Example graphs and their adjacency matrices. The triangle feature of all the three graphs could be squared in their adjacency matrices.

首先对比5(a)和5(b)，与图5(a)相比，图5(b)加入了节点1，节点顺序依次后移一位，但三角形特征中三个节点的顺序还是连续的，这个特征在邻接矩阵中可以看到，这个1-1-0-1的2x2矩阵仍然存在。

CNN能捕捉这些信息： CNN在训练集中看过很多这种样例后，可以学习到这种平移不变性。

再看图5(c)，其加入的节点2打破了原有三角形的节点顺序，然而，在邻接矩阵中我们可以看到它实际上是把原来的2x2矩阵放大成了3x3矩阵，当我们移除第二行和第二列时，仍然可以得到一个1-1-0-1的2x2矩阵。

CNN能学习伸缩不变性的信息： 这种这与图像中的image scaling类似，CNN在训练集中包含足够多样例的情况下，也是可以学到这种伸缩不变性的。

我们讨论了CNN的平移不变性和尺度不变性可以学习节点顺序的小变化。在二进制代码相似性检测任务中，当在不同的平台上编译相同的函数时，节点的顺序通常没有很大的变化。大多数节点顺序的变化都是添加一个节点、删除一个节点或交换几个节点，所以CNN在我们的任务中很有用。除了提高学习节点顺序信息的准确性外，CNN还有几个额外的优点。

- 首先，与传统的图特征提取算法相比，直接在邻接矩阵上使用CNN要快得多。
- 其次，CNN可以添加到不同大小的输入上，因此它可以建模不同大小的图形，而不需要进行填充和剪切等预处理

> **原文：**
> Most node order changes are adding a node, deleting a node, or exchanging several nodes, so CNN is useful on our task. Except the accuracy improvement on learning node order information, CNN has several additional advantages. First, comparing with traditional graph feature extracting algorithms, using CNN directly on adjacency matrices is much faster. Second, CNN could be added on inputs with different sizes, so it could model different-size graphs without pre-processing such as padding and clipping.

我们在任务中使用Resnet（He et al. 2016），使用一个11层的Resnet，包含3个residual block，所有的feature map大小均为3x3。之后用一个global max pooling层，得到graph order embedding。在此之前不用pooling层，因为输入的图的大小不同。具体如公式8所示。

$$g_o = \text{Maxpooling}(\text{Resnet}(A)) \tag{8}$$

五.实验分析

1.Datasets

本文在两个任务上进行实验。

- 任务1是跨平台二进制代码分析，同一份源代码在不同的平台上（x86-64和ARM）进行编译，我们的目标是使模型对同一份源代码在

不同平台上编译的两个控制流图pair的相似度得分高于不同源代码pair的相似度得分。

- 任务2是二进制代码分类，判断控制流图属于哪个优化选项（O2和O3）。

请注意，我们的方法对于检测不同编译器（如clang & gcc）上的二进制代码也很有用，在本文中，我们不选择它作为数据集。数据集的基本统计数据如表1所示。任务1是排序问题，因此使用MRR10和Rank1作为评价指标。任务2是分类问题，因此使用准确率作为评价指标。

Dataset	Task	Training	Validation	Testing
gcc-O2	1	31,410	3,857	3,884
gcc-O3	1	16,059	4,155	4,077
gcc-x86-64	2	27,761	3,406	3,492
gcc-ARM	2	9,447	4,773	4,933

Table 1: Basic statistics of the datasets.

2.Compared Methods

因为我们的模型有三个组成部分，所以进行了不同的实验来找出每个部分的影响。

- Graph kernel methods
- Gemini
- MPNN
- Word2Vec
- Skip thought
- BERT
- CNN-based models
- CNN (random)
- MPNN (without semantic)
- MPNN (without semantic) + CNN
- Our model：Our model is BERT (4 tasks) + MPNN + 11-layer Resnet, which contains both semantic-aware modeling, structural-aware modeling, and order-aware modeling.

3.Results

总体性能（Overall performance）

表2和表3显示了不同模型在两个任务上的总体性能。表中第一个分块是整体模型，包括graph kernel，Gemini以及MPNN模型。第二个分块是semantic-aware模块的对比实验，分别使用了word2vec，skip thought，以及BERT，其中BERT2是指原始BERT论文中的两个task（即MLM和ANP），BERT4是指在此基础上加入两个graph-level task（BIG和GC）。第三个分块是对order-aware模块的对比实验，基础CNN模型使用3层CNN以及7、11层的Resnet，CNN_random是对训练集中控制流图的节点顺序随机打乱再进行训练，MPNN_ws是去除控制流图节点中的语义信息（所有block向量设为相同的值）再用MPNN训练。最后是本文的最终模型，即BERT+MPNN+Resnet。

Model	Task1-O2	Task1-O3
Weisfeiler-Lehman	0.2493 / 0.1810	0.1940 / 0.1565
Gemini	0.6069 / 0.5491	0.5430 / 0.4760
MPNN	0.6096 / 0.5507	0.5501 / 0.4802
word2vec	0.7003 / 0.6534	0.6198 / 0.5555
skip thought	0.6825 / 0.6238	0.5954 / 0.5226
BERT2	0.7591 / 0.7060	0.6507 / 0.5852
BERT4	0.7704 / 0.7233	0.6672 / 0.5989
CNN3 (random)	0.0362 / 0.0020	0.0307 / 0.0015
CNN3	0.4142 / 0.3684	0.3074 / 0.2612
Resnet7	0.4330 / 0.3868	0.3229 / 0.2732
Resnet11	0.4419 / 0.3952	0.3271 / 0.2837
MPNN_{ws}	0.3361 / 0.3014	0.2161 / 0.1913
MPNN_{ws}+Resnet11	0.4457 / 0.3970	0.3348 / 0.2907
Our model	**0.7922 / 0.7421**	**0.6855 / 0.6114**

Table 2: Performance comparison of task 1 in terms of MRR10 / Rank1.

本文提出的模型与Gemini模型相比，在任务1和任务2上的评价指标分数均大幅提升。semantic-aware模块使用NLP模型（word2vec，BERT等）均优于使用人工提取的特征。只使用order-aware时模型也取得了不错的效果。与其它所有模型相比，本文提出的模型均取得了更优的效果。

Model	Task2-x86-64	Task2-ARM
Weisfeiler-Lehman	-	-
Gemini	77.88	79.89
MPNN	79.65	80.62
word2vec	82.24	84.23
skip thought	80.43	83.74
BERT2	82.67	85.19
BERT4	83.74	86.33
CNN3 (random)	66.06	64.57
CNN3	82.11	83.70
Resnet7	82.56	84.13
Resnet11	82.64	84.24
MPNN_{ws}	76.29	76.90
MPNN_{ws}+Resnet11	82.92	85.05
Our model	**86.14**	**88.41**

Table 3: Performance comparison of task 2 in terms of accuracy (%).

语义感知（Model variants for semantic-aware modeling）

只看表中第二个分块，BERT的结果优于word2vec和skip thought，因为BERT能在预训练过程中提取更多的信息。 为了验证BERT预训练是否必要和有效，我们研究了几个变体。首先，基于NLP的训练前块特征（word2vec、skip thought、BERT 2和4）比手动特征具有更好的性能，这表明为CFG块构建复杂的模型是必不可少的。与word2vec和skip thought相比，使用MLM和ANP任务的BERT不仅考虑块级预测，还考虑令牌级预测，并且双向转换器更具有提取有用信息的能力。

BIG任务和GC任务也很有用，其结果增加了1% - 2%。在这两个任务中，块嵌入可以学习图信息，这可能有助于处理图任务。在图6中显示了块嵌入，四个cfg及其块嵌入被设置在四个方向上。我们采用K-means将这些块嵌入聚为四类，不同的集群有不同的颜色（红色、蓝色、绿色和紫色）。我们可以观察到，同一图中的块的趋势是有相同的颜色，而不同的图有不同的主色。

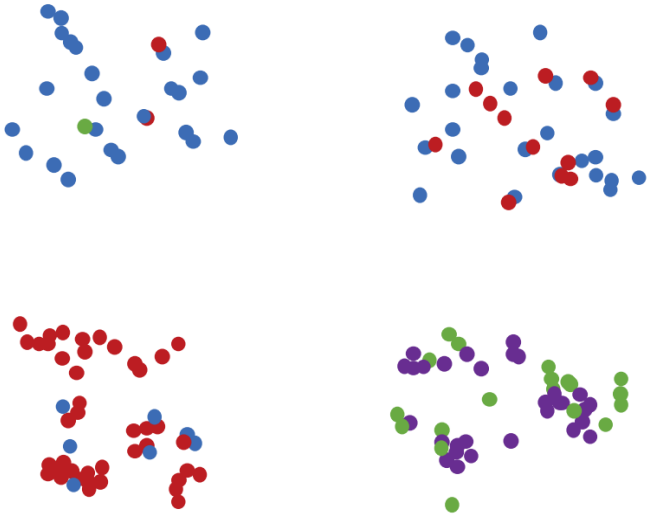


Figure 6: Visualization of 4 CFGs and their block embeddings. Different classes clustered by K-means have different colors.

顺序感知（Model variants for order-aware modeling）

只有使用基于cnn的模型才能在这两个任务上都能取得良好的效果。11层Resnet比3层CNN和7层Resnet稍微好一些。与mpnws相比，基于cnn的模型获得了更好的性能。当随机变换节点时，CNN什么也学不到。这意味着CNN模型可以学习节点的顺序

观察表中第三个分块，CNN模型在两个任务上都取得了不错的效果。Resnet11优于Resnet7和CNN3。与MPNN_ws相比，CNN效果更优。随机打乱节点顺序后，CNN模型效果大幅下降，这表示CNN模型确实可以学到节点顺序信息。图7是控制流图pair的例子，这个函数为“ZN12libfwbuilder15RuleElementRGtw13validateC-hildEPNS8FWObjectE“。

- 左边是在gcc&x86-86上编译的控制流图
- 右边是在gcc&ARM上编译的控制流图

可以看到，左图的节点3在右图中被拆成节点3和节点4，除此之外其它节点的顺序与边的连接方式均相同。经过CNN模型的计算，这两个图的cosine相似度为0.971，排序rank的排名为1。这表明CNN模型可以从邻接矩阵中学到控制流图的节点顺序。

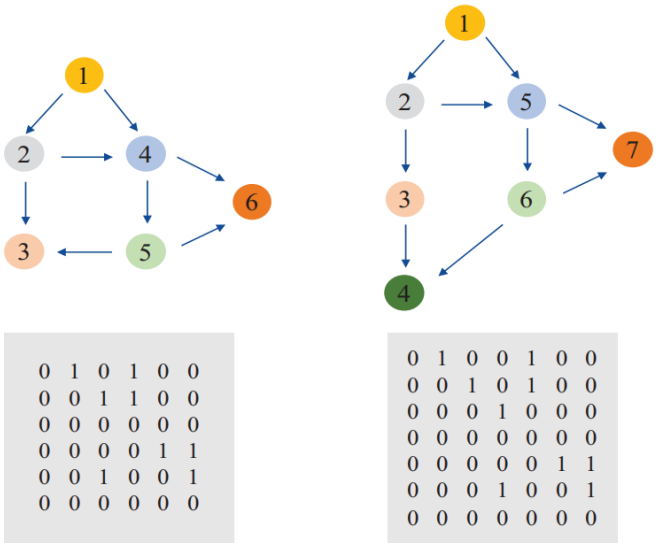


Figure 7: An example of CFG changes, these two CFGs' cosine similarity score is 0.971.

六.结论

本文提出了一个新的模型，用于解决二进制代码分析的问题。本文的模型中包含semantic-aware模块，structural-aware模块以及order-aware模块。我们观察到语义信息和节点顺序信息都是控制流图重要的特征。我们使用BERT预训练模型提取语义信息，并使用CNN模型提取节点顺序信息。实验结果表明，本文提出的模型与之前最优的模型相比，取得了更好的效果。

七.个人感受

这篇文章就写到这里，希望对您有所帮助。由于作者英语实在太差，论文的水平也很低，写得不好的地方还请海涵和批评。同时，也欢迎大家讨论，继续加油！感恩遇见，且看且珍惜。

(By:Eastmount 2023-04-06 周四夜于武汉 http://blog.csdn.net/eastmount/)