**传统的：**

Trex（2020） 是一种最先进的BCSD方法，它使用基于微跟踪的基于转移学习的模型来生成用来匹配类似函数的函数嵌入

https://github.com/CUMLSec/trex

《Trex: Learning execution semantics from micro-traces for binary similarity》

**基于深度学习的：**

SAFE （2019）是一种基于注意力的汇编语言嵌入模型。它使用了一个带有注意机制的RNN架构来生成函数嵌入。我们使用他们的官方开源源代码和预先训练的模型进行默认参数评估。<https://github.com/facebookresearch/SAFEtorch>

《Safe: Self-attentive function embeddings for binary similarity》

Iv2\_attention（2019）这个基线使用word2vec 来学习指令标记的嵌入。接下来，它使用一个RNN来为每个基本块生成独立的嵌入，最后使用strutc2vec来组合嵌入并生成所分析的函数的表示。https://github.com/lucamassarelli/Unsupervised-Features-Learning-For-BinarySimilarity

《Investigating graph embedding neural networks with unsupervised features extraction for binary analysis》

**基于自然语言的：**

Asm2vec（2019）是一个基于pv-dm的用于汇编语言嵌入的模型。它在CFG上使用随机游动来对指令序列进行采样，然后使用PV-DM模型来学习汇编语言的嵌入。Asm2vec不是开源的，我们使用了一个公开可用的非官方版本的。我们使用了默认的参数来进行评估。https://github.com/oalieno/asm2vec-pytorch

《Asm2vec: Boosting static representation robustness for binary clone search against code

obfuscation and compiler optimization》

PalmTree（2021）是一种基于bert的装配指令嵌入模型。它使用了控制流与数据流和三个预训练任务来学习装配指令的特征，并生成指令嵌入。我们基于他们的官方开放源代码和具有默认参数的预训练模型实现了这个基线。

https://github.com/palmtreemodel/PalmTree

《PalmTree: Learning an Assembly Language Model for Instruction Embedding》

**基于图神经网络的：**

Gemini（2017）这个基线为每个基本块提取手工制作的特征，并使用GNN来学习所分析的函数的CFG表示。我们基于其官方的张量流代码实现了这种方法，并在整个评估过程中使用了它的默认参数设置.

https://github.com/xiaojunxu/dnn-binary-code-similarity

《Neural network-based graph embedding for cross-platform binary code similarity detection》

字符串No. of string constants

立即数No. of numeric constants

传输指令No. of transfer instructions

跳转指令No. of calls instructions

算术指令No. of arithmetic instructions

一个基本块的指令个数No. of total instructions

节点的图中心性Betweenness: