



Python Bindings for IR2Vec

Nishant Sachdeva
S. VenkataKeerthy

Prof. Ramakrishna Upadrasta
Scalable Compilers for Heterogeneous
Architectures Group, IIT Hyderabad

EuroLLVM 2026

Background

- ML guided optimizations in LLVM
- MLGO - Inlining for size, Register Allocation

Features	Embeddings
Handcrafted (counts, depths, CFG metrics)	Learned automatically
Task-specific	General-purpose, reusable

IR2Vec



- LLVM IR based embeddings
- IR2Vec with MLGO for inlining yields up to 5% code size reduction
 - over -Os *
- Shown effectiveness on different ML-driven optimizations

RL-Loop Distribution

Jain, VenkataKeerthy, et al,
LLVM HPC'22

Phase Ordering (POSET-RL)

Jain, VenkataKeerthy, et al,
ISPASS'22

Register Allocation (RL4ReAL)

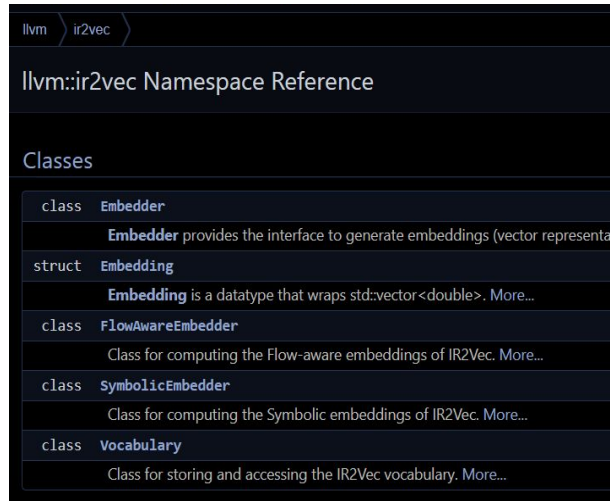
VenkataKeerthy, Jain, et al,
CC'23

- <https://discourse.llvm.org/t/rfc-enhancing-mlgo-inlining-with-ir2vec-embeddings/86250>

Current ecosystem

- IR2Vec in `llvm/Analysis`
- MIR2Vec in `llvm/CodeGen`
- `llvm-ir2vec` standalone tool in `llvm/tools`
 - Supports embedding generation

```
proc = subprocess.run([
    "llvm-ir2vec", "embeddings",
    "--ir2vec-vocab-path=vocab.json",
    "--ir2vec-kind=flow-aware",
    ll_file
], capture_output=True, text=True)
embeddings = parse_ir2vec_output(proc.stdout)
```




llvm ir2vec

llvm::ir2vec Namespace Reference

Classes

class	Embedder	Embedder provides the interface to generate embeddings (vector representa...
struct	Embedding	Embedding is a datatype that wraps std::vector<double>. More...
class	FlowAwareEmbedder	Class for computing the Flow-aware embeddings of IR2Vec. More...
class	SymbolicEmbedder	Class for computing the Symbolic embeddings of IR2Vec. More...
class	Vocabulary	Class for storing and accessing the IR2Vec vocabulary. More...



 **LLVM**
C O M P I L E R
I N F R A S T R U C T U R E

[LLVM Home](#) | [Documentation](#) » [Reference](#) » [LLVM Command Guide](#) » [llvm-ir2vec - IR2Vec and MIR2Vec Embedding Generation Tool](#)

llvm-ir2vec - IR2Vec and MIR2Vec Embedding Generation Tool

SYNOPSIS

```
llvm-ir2vec [subcommand] [options]
```

DESCRIPTION

llvm-ir2vec is a standalone command-line tool for IR2Vec and MIR2Vec. It generates embeddings for both LLVM IR and Machine IR. It also supports triplet generation for vocabulary training.

The tool provides three main subcommands:

- triplets**: Generates numeric triplets in train2id format for vocabulary training from LLVM IR.
- entities**: Generates entity mapping files (entity2id.txt) for vocabulary training.
- embeddings**: Generates IR2Vec or MIR2Vec embeddings using a trained vocabulary at different granularity levels (instruction/function).

The tool supports two operation modes:

Proposed `ir2vec` python package

- Uses `nanobind`
- Available upstream
- Currently, wheels built and published downstream to testPypi as `llvm-ir2vec`

```
pip install -i https://test.pypi.org/simple/ llvm-ir2vec
```

- Wheels come bundled with pre-trained vocabulary files
 - 75D, 100D, 300D

`ir2vec` Python API



```
import ir2vec

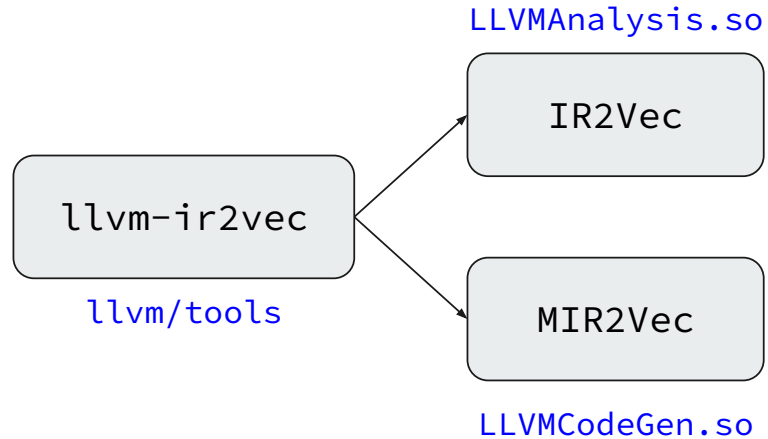
vocabObj = ir2vec.loadVocab(
    ir2vec.vocab.seedEmbedding75D
)

emb = ir2vec.initEmbedding (
    filename="file_path.ll",
    mode=ir2vec.IR2VecKind.FlowAware,
    vocab=vocabObj
)
```

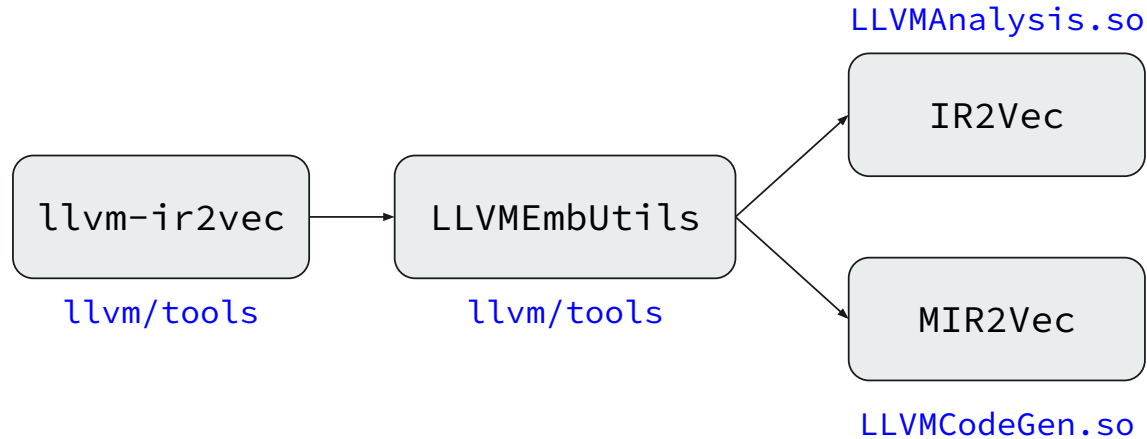
```
func_names = emb.getFuncNames()
func_emb_map = emb.getFuncEmbMap()

# for an IR function "foo"
func_emb = emb.getFuncEmb("foo")
bb_map = emb.getBBEmbMap("foo")
inst_map = emb.getInstEmbMap("foo")
```

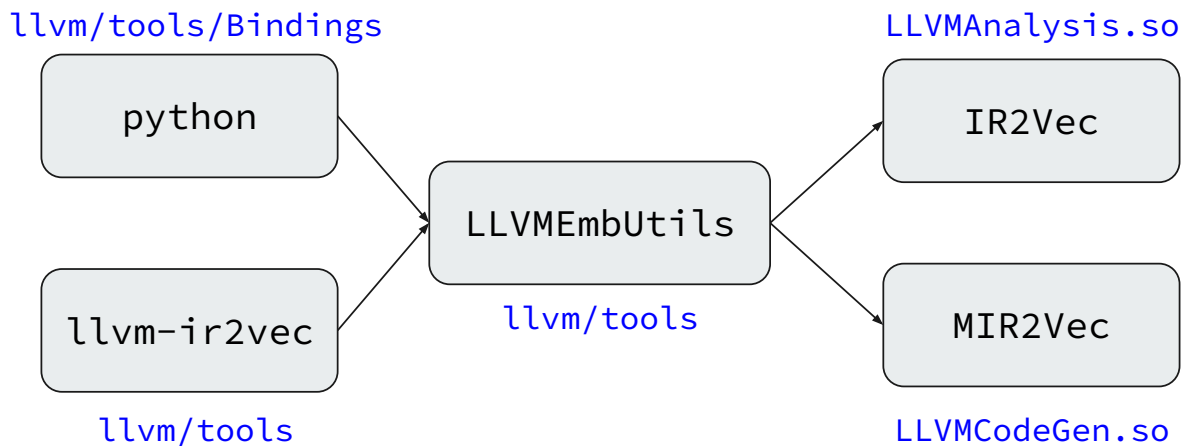
Implementation specifics - Python Bindings



Implementation specifics - Python Bindings



Implementation specifics - Python Bindings



Wins ? - Performance

- Scalability stress test
 - Given a dataset
 - Collect function embedding maps and record the time-taken

```
@timed()
def benchmark(ll_files):
    . . .
    for ll_file in ll_files:
        proc = subprocess.run(...)
        embeddings = parse_output(proc.stdout)
        embedding_dict[ll_file] = embeddings
```

llvm-ir2vec

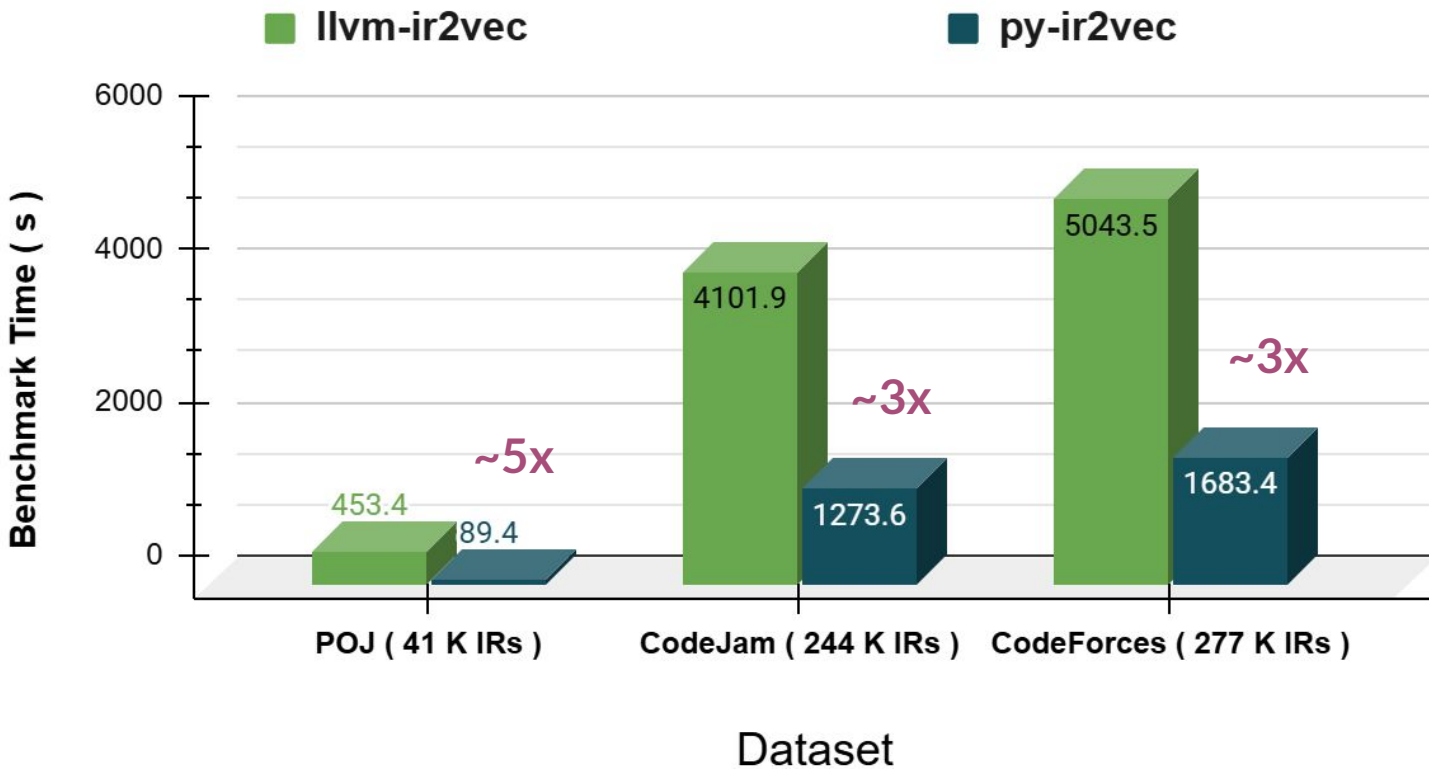
Wins ? - Performance

- Scalability stress test
 - Given a dataset
 - Collect function embedding maps and record the time-taken

```
@timed()
def benchmark(ll_files):
    . . .
    vocabObj = ir2vec.loadVocab(...)
    for ll_file in ll_files:
        tool = ir2vec.initEmbedding(filename, mode, vocab)
        embeddings = tool.getFuncEmbMap()
        embedding_dict[ll_file] = embeddings
```

py-ir2vec

Benchmark Function Embedding collection from a dataset



Pain points addressed



- Subprocesses removed
- Output parsing bypassed
- Easy Error Handling
- Speed enhancements

Next Steps

- Currently, a downstream repo builds and publishes `llvm-ir2vec` wheels to TestPyPi
- Distribution
 - `ir2vec` wheels built and published as part of LLVM
 - No separate repo, No maintenance drift

Artifacts for this work are available at



- Technical Report - [HERE](#)
- Colab Demo - [HERE](#)

Acknowledgements



Scalable Compilers for Heterogeneous Architectures Group, IIT Hyderabad

- S. VenkataKeerthy
- Prof. Ramakrishna Upadrasta



Thank You