Making use of knowledge graphs for code

Kavitha Srinivas, IBM Research

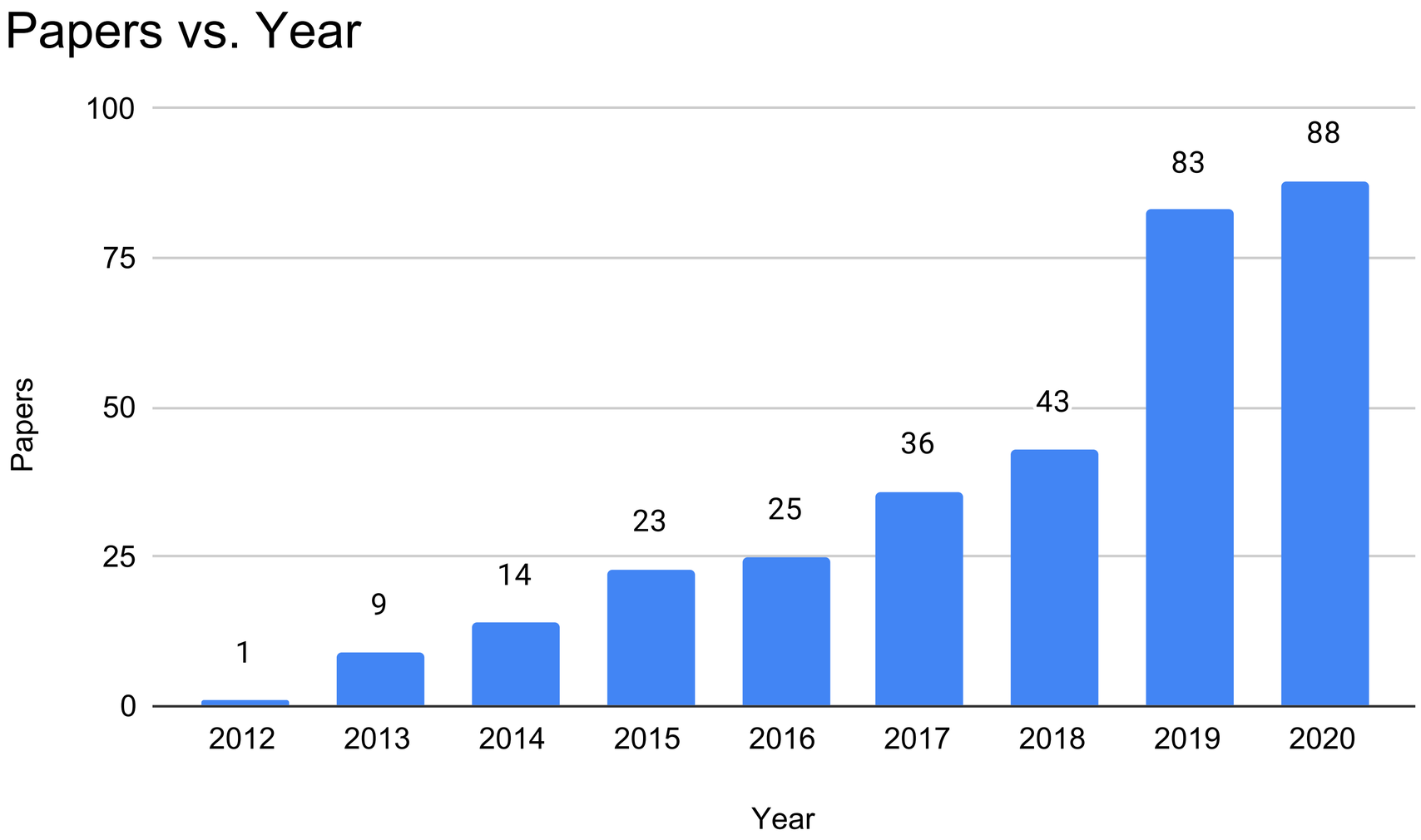
Outline

- Knowledge graphs for code - Why build it?
- **Graph4Code** - design principles
- Challenges in using the graph for knowledge infused learning

Increased interest in AI for code

Machine learning for code

Increasing interest in applying machine learning for code



Source: https://ml4code.github.io/

Neural models are being used for better...

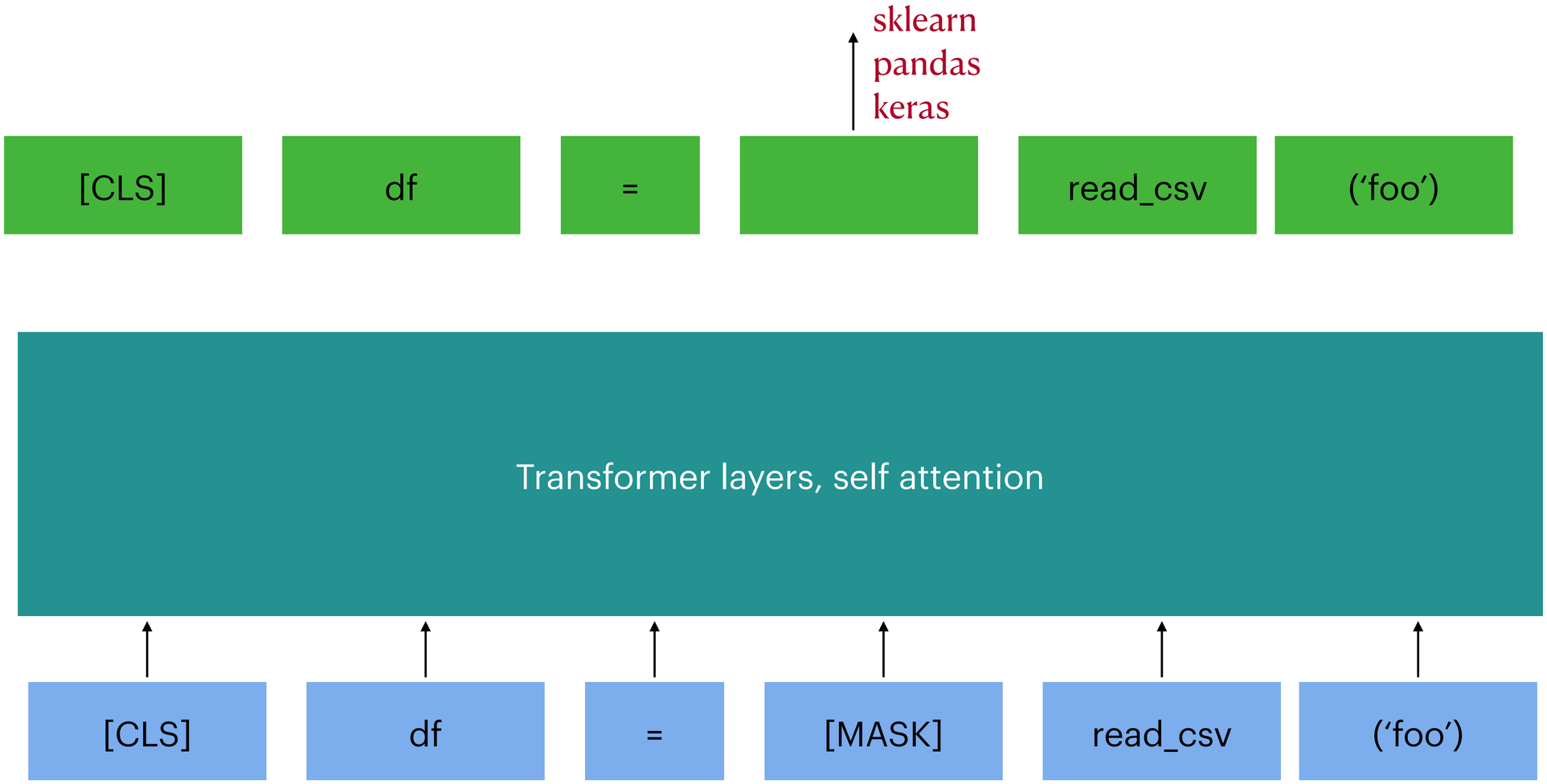
- Code search
- Clone detection
- Code refactoring
- Bug detection
- Vulnerability analysis
- Code recommendations
- Enforcing coding best practices

.....

Heavy use of neural modeling approaches from **natural language** to code

Language models for code

Example systems - cuBERT, CodeBERT, TransCoder



df = pandas.read_csv('foo')

What objective?
- Masked language
- Denoising autoencoder
- Replaced token

What tokens?
-Language specific
-Natural language

What scope?
- Line
- Function

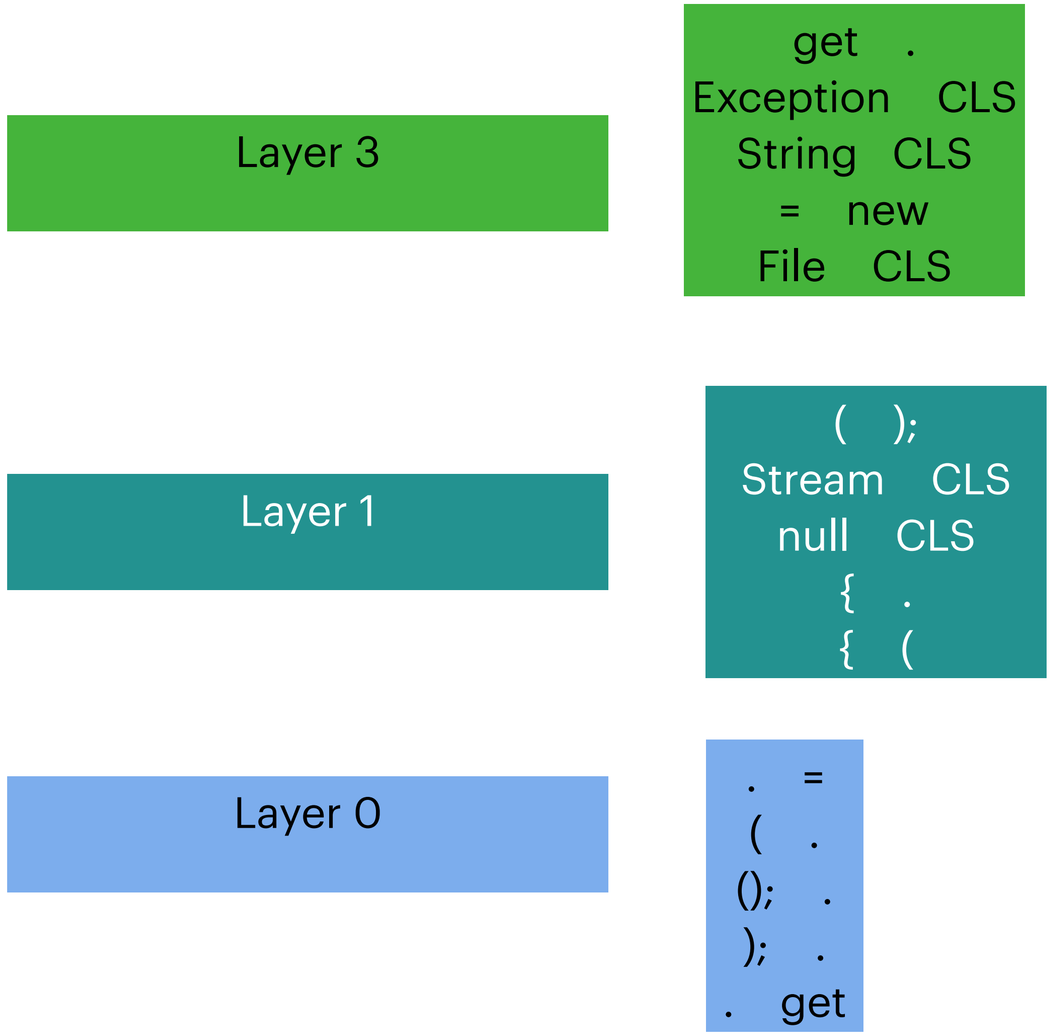
Code vs. natural language

Code has unique attributes compared to natural languages

- **abstract:** rename all variables and it still has the same semantics
- **non-local:** program flow spans many different lines of code, and even across documents.

Use of natural language tokens, single program lines (or even functions) limits the system's capability to learn the semantics of code.

Language models and syntactic learning



Given program size constraints, and token types used, models learn very **local** properties of code - nothing about program scopes (e.g., matching parens), let alone function calls.

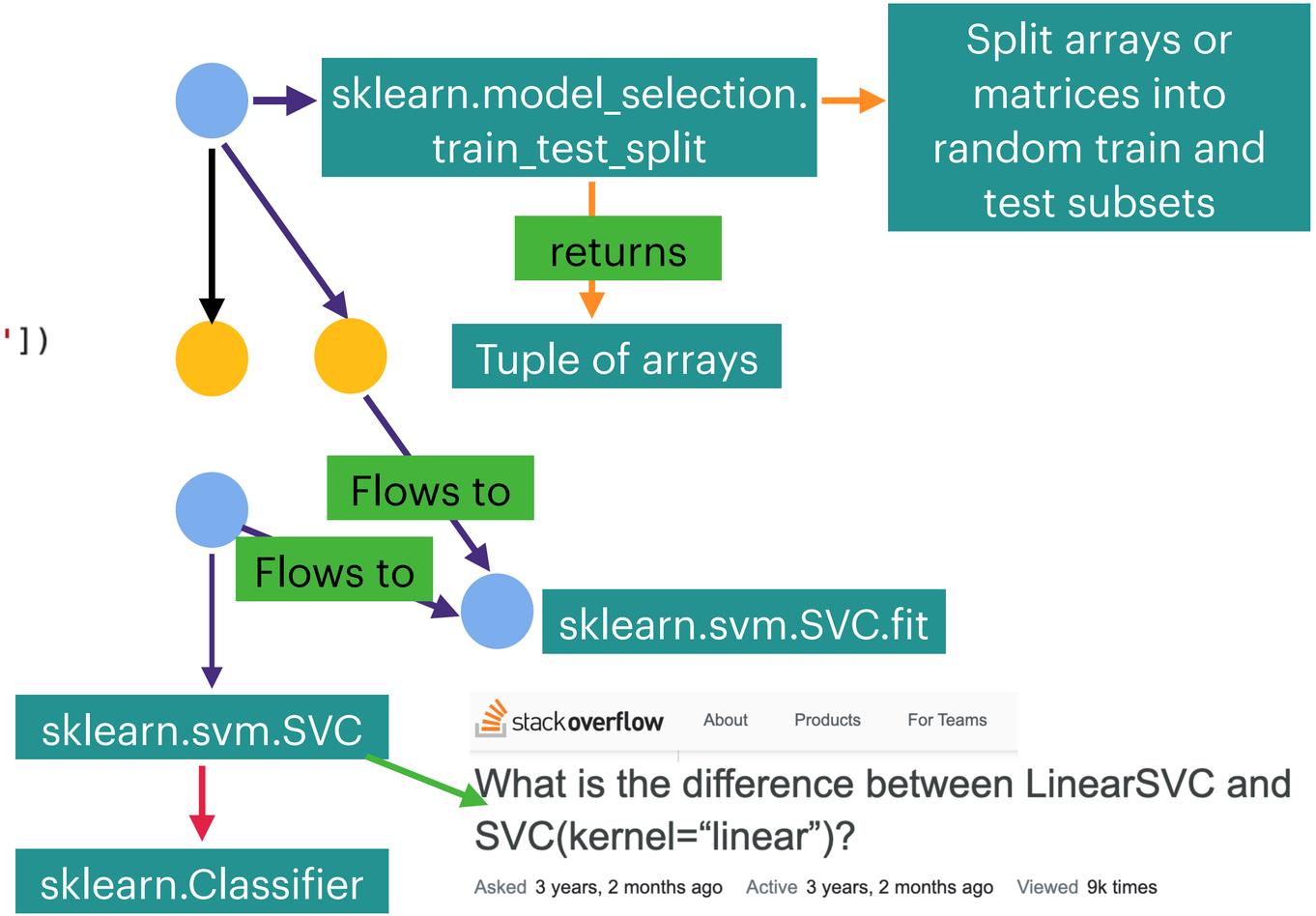
Analysis from CodeBERT: top token pairs attended to by layer show **syntactic** learning. Very short range dependencies (e.g., ( )).

CuBERT models not different.

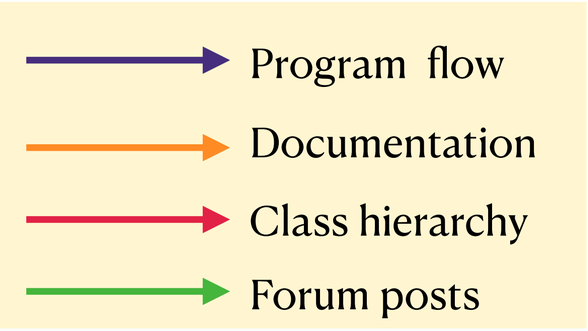
How do we teach systems **semantics** of programs?

How do we as humans understand code semantics?

```python
# In[109]:


train, test = train_test_split(my_df,
                               test_size = 0.3,
                               random_state = 0,
                               stratify = my_df['Dataset'])
train_X = train[train.columns[:len(train.columns)-1]]
test_X = test[test.columns[:len(test.columns)-1]]
train_Y = train['Dataset']
test_Y = test['Dataset']


# In[113]:


types=['rbf','linear', 'sigmoid']
for i in types:
    model = svm.SVC(kernel=i, random_state=0)
    model.fit(train_X,train_Y)
```



Code semantics is buried in multiple sources
Can we build a knowledge graph for code to capture program semantics?



Outline

- Knowledge graphs for code - Why build it?
- **Graph4Code - design principles**
- Challenges in using the graph for knowledge infused learning

Graph4Code Construction

Gather semantics by program flow: Static or dynamic?

1M+ GitHub Python scripts

```
1 data = pd.read_csv("../input/indian_liver_patient.csv", low_memory=False)
2 data = data.where(pd.notnull(data), data.median(), axis='columns')
3 train, test = train_test_split(my_df,
                                 test_size = 0.3,
                                 random_state = 0,
                                 stratify = my_df['Dataset'])
4 train_X = train[train.columns[:len(train.columns)-1]]
```

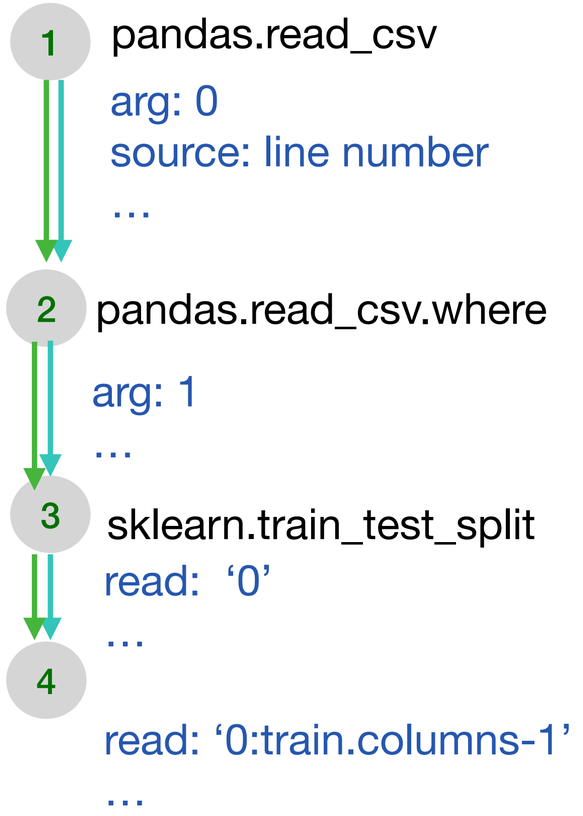
immediatelyPrecedes

flowsTo

edge annotations on flowsTo

arg, source, read…

One named graph per program



Static Analysis Challenges

```
# In[109]:


train, test = train_test_split(my_df,
                               test_size = 0.3,
                               random_state = 0,
                               stratify = my_df['Dataset'])
train_X = train[train.columns[:len(train.columns)-1]]
test_X = test[test.columns[:len(test.columns)-1]]
train_Y = train['Dataset']
test_Y = test['Dataset']


# In[113]:


types=['rbf','linear', 'sigmoid']
for i in types:
    model = svm.SVC(kernel=i, random_state=0)
    model.fit(train_X,train_Y)
```

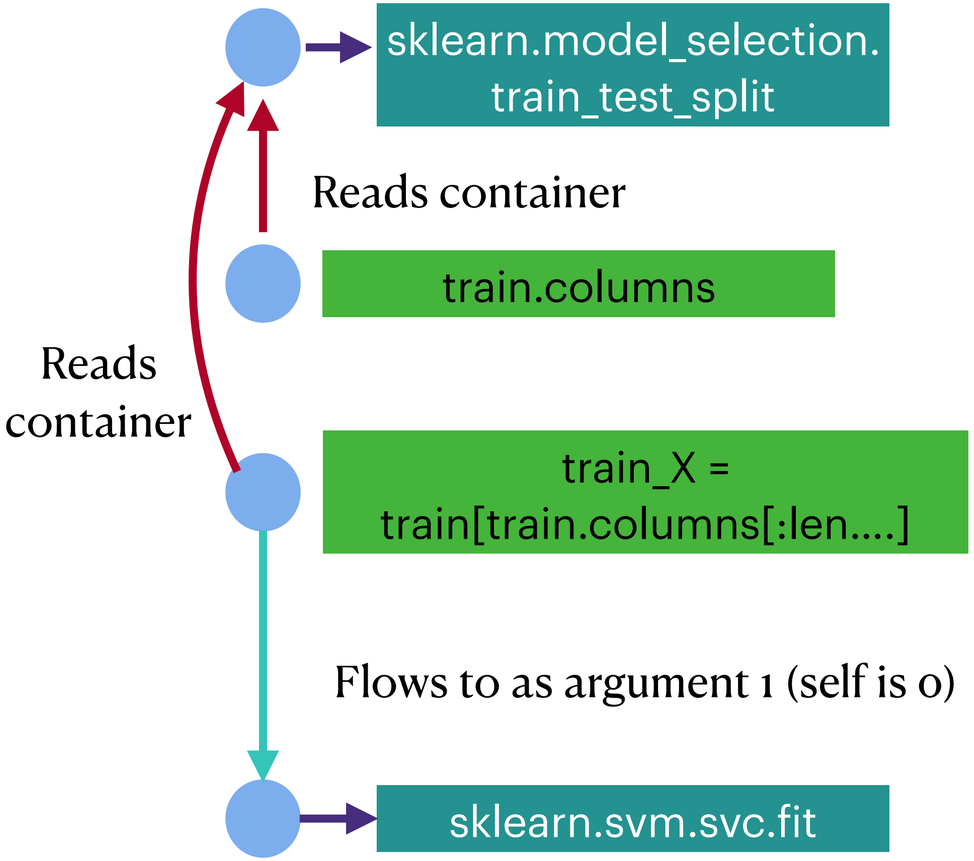
Heavy use of libraries, source often not possible to analyze (not even Python)

Code needs to model heap structures (first element of tuple flows into fit call)

Modified the WALA libraries to be able to analyze vast number of programs efficiently, using abstractions for library calls and field accesses

Static analysis graphs of code

```
# In[109]:


train, test = train_test_split(my_df,
                               test_size = 0.3,
                               random_state = 0,
                               stratify = my_df['Dataset'])
train_X = train[train.columns[:len(train.columns)-1]]
test_X = test[test.columns[:len(test.columns)-1]]
train_Y = train['Dataset']
test_Y = test['Dataset']


# In[113]:


types=['rbf','linear', 'sigmoid']
for i in types:
    model = svm.SVC(kernel=i, random_state=0)
    model.fit(train_X,train_Y)
```



A subgraph

Documentation/class hierarchies

Identify the top libraries (based on usage)

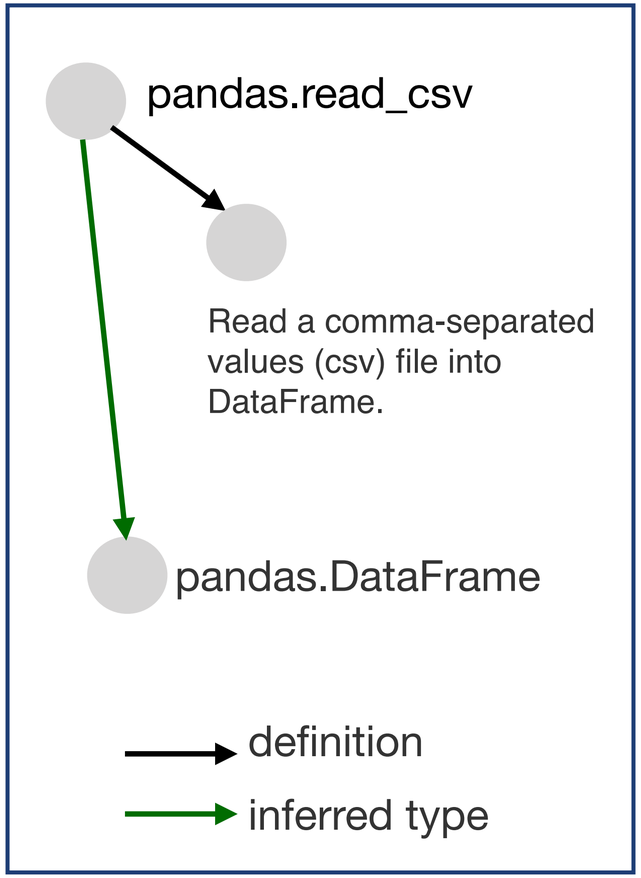
Dynamically install module, inspect module

Gather documentation, class hierarchies

Parse possible return and parameter types using regex

Use information retrieval on class names to infer possible return or parameter types

6M+ functions, classes, methods



Connecting to forum posts

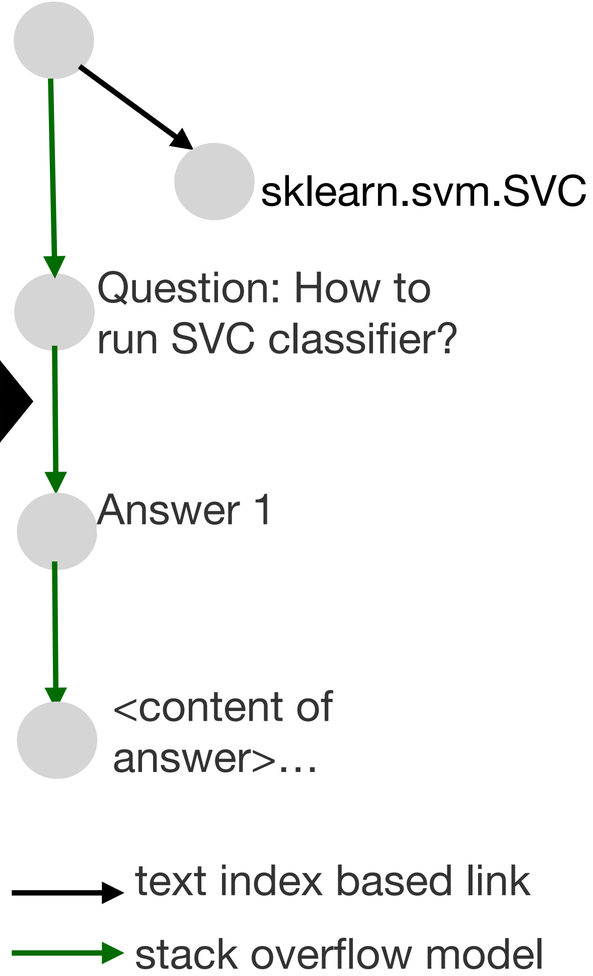
47M forum posts

Represent questions, answers as a single document

Use text index to index the text (custom analyzer for code)

Find all mentions to classes, functions, methods using IR techniques

Structure post into question, answer, code snippets etc.



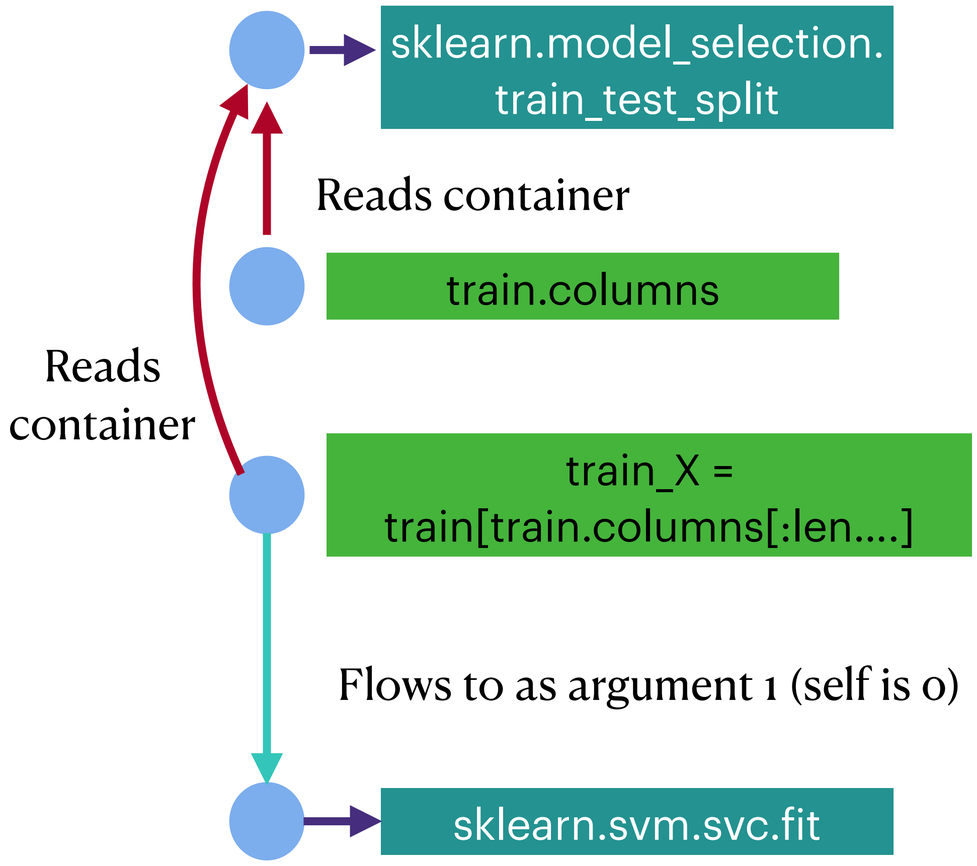
Graph4Code Statistics

Sources	Classes	Methods	Functions
Inspection	5.8M	257K	278K
Web forums	88K	742K	106K
Program flow	2.1M	959K	4.2M

Outline

- Knowledge graphs for code - Why build it?
- **Graph4Code** - design principles
- **Challenges in using the graph for knowledge infused learning**

Can we build better models for code?



A subgraph

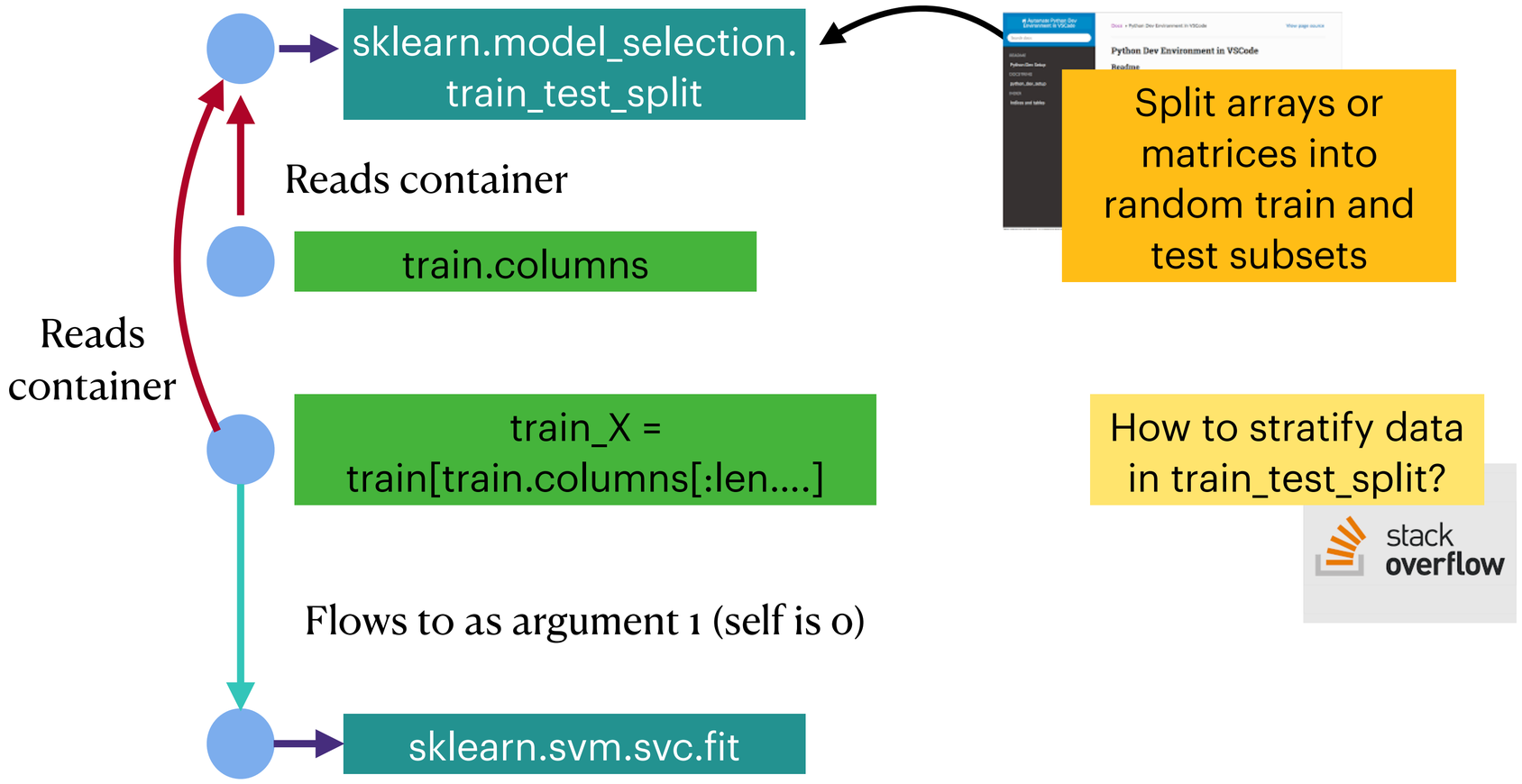
How to build node embeddings, independent of call stack? E.g., Python's lack of type system leads to proliferation of node types

How to build GNNs across multiple edge types?

How to handle directionality in propagating across the network?

Is graph structure key to learning or paths in the program a better representation?

And that isn't even including the rich semantics of code



A subgraph

How do we capture the text around code artifacts into useful node embeddings?

Additional Details

https://wala.github.io/graph4code/
2 B triples graph
Version 2 which is a richer representation of the graph coming soon!