SOAR: Synthesis for Open-Source API Refactoring

Aidan Z.H. Yang

Queen's University
Carnegie Mellon University
Motivation

• As software evolves, the API calls it depends on may become obsolete [1].

• No existing work that takes an enumerative program synthesis approach towards API refactoring.

• Existing work learns and recommends potential API mappings [1,2], or migrates APIs based on a large dataset of existing migrations [3].

Motivation

A: tf.Conv2d parameter filters has no corresponding name parameter.

B: tf.Dense has no torch function of the same name.

C: Two functions from torch are required to perform the same task as tf.Conv2DTranspose.
Key Insights

Documentation on data science APIs have description on parameters, as well as call examples.

A `tf.Tensor` has the following properties:

- a data type (float32, int32, or string, for example)
- a shape

Each element in the Tensor has the same data type, and the data type is always known.

In eager execution, which is the default mode in TensorFlow, results are calculated immediately.

```python
>>> # Compute some values using a Tensor
>>> c = tf.constant([[1.0, 2.0], [3.0, 4.0]])
>>> d = tf.constant([[1.0, 1.0], [8.0, 1.0]])
>>> e = tf.matmul(c, d)
>>> print(e)
tf.Tensor(

[1. 3.]

[8. 7.]], shape=(2, 2), dtype=float32)
```

**Parameters**

- `in_channels (int)` - Number of channels in the input image
- `out_channels (int)` - Number of channels produced by the convolution
- `kernel_size (int or tuple)` - Size of the convolving kernel
- `stride (int or tuple, optional)` - Stride of the convolution. Default: 1
- `padding (int or tuple, optional)` - Zero-padding added to both sides of the input. Default: 0
- `padding_mode (string, optional)` - 'zeros', 'reflect', 'replicate' or 'circular'. Default: 'zeros'
- `dilation (int or tuple, optional)` - Spacing between kernel elements. Default: 1
- `groups (int, optional)` - Number of blocked connections from input channels to output channels. Default: 1
- `bias (bool, optional)` - If True, adds a learnable bias to the output. Default: True
Key Insights

Error messages contain information on the input matrix.

- Trying to create tensor with negative dimension -2: [40, -2, 3, 3].
- `embedding()`: argument weight (position 1) must be Tensor, not int.
- Expected 3-dimensional input for 3-dimensional weight [2, 2, 3], but got 4-dimensional input of size [100, 50, 40, 1] instead.
General Framework

Input:
- A program written with source API.
- Documentation of APIs.
- Some test cases for the original program.

Output:
- Program with the same functionality written with the target library.
API Matching

Output:
• A probability distribution over all target APIs
  -> Top 10 most similar APIs by cosine-similarity.

Model:
• API representation learning, represent each API in a continuous vector space.
• Learned with the API documentation (page title, code example etc…) and GloVe pre-trained NLP data.

\[
\text{Embedding}(x^i) = \sum_{i=j}^{n} \frac{x^i_j \cdot w_j}{\sum_{i=0}^{m} x^i_j}
\]

\[
\text{sim}(\text{Rep}(x^i), \text{Rep}(x^j)) = \frac{\text{Rep}(x^i) \cdot \text{Rep}(x^j)}{|\text{Rep}(x^i)||\text{Rep}(x^j)|}
\]
Program Synthesis

**Algorithm 1 SYNTHESIZER(I, S, T, C)**

*Input:* I: existing program, S: source library, T: target library, C: test cases

*Output:* O: refactored program

1. \( \tilde{r} : \text{API mapping} = \text{MAPAPI}(T, S) \)
2. \( O = \{ \} \)
3. \textbf{for each} \( l \in I \) \textbf{do}
4. \( O = O \cup \text{REFACTORLINE}(l, T, C, \tilde{r}) \)
5. \textbf{end for}

**Algorithm 2 REFACTORLINE(l, T, C, \tilde{r})**

*Input:* l: line of code from I, T: target library, C: test cases, \( \tilde{r} \): ranked list of API matchings

*Output:* \( \mathcal{R} \): refactored snippet

1. \( O = \{ \} \)
2. \textbf{for each} \( l' \in \tilde{r} \) \textbf{do}
3. \( \tilde{s} = \text{GENERATESKETCHES}(l', T) \)
4. \textbf{for each} \( s \in \tilde{s} \) \textbf{do}
5. \( \mathcal{R} = \text{FILLSKETCH}(s) \)
6. \textbf{if} \( \text{PASSTESTS}\mathcal{R}, C \text{ then} \)
7. \( \text{return } \mathcal{R} \)
8. \textbf{end if}
9. \textbf{end for}
10. \textbf{end for}

**Program Sketching:**
Sketch a new program with holes based on the original program template.

**Program Enumeration**
Enumerate through all likely target calls with all likely parameters.
Pruning the Search Space

1. Specification inferred from documentation.
   - `in_channels (int)` – Number of channels in the input image

2. Constraints from Error messages.
Error Message Understanding

- Generate SMT constraints from run-time errors.
Example Pipeline
Results

- Collected the top 1015 starred repositories that have TensorFlow as a topic tag.
- Over 8 million lines of code and over 500K TensorFlow API calls.
- 76% of the repositories use API calls included in our benchmarks at least once.
Conclusion

• SOAR is the first to use NLP and synthesis for automatic API refactoring.

• SOAR requires no existing migration as training data.

• SOAR can successfully migrate 80% of neural network programs composed by 3 to 44 layers in with an average time of 97.23 seconds.

• For Dplyr to Pandas migration, 90% of benchmarks are solved on average in 17.31 seconds.

• With ablation studies, we show that the use of specifications from API documents and learning from error messages can improve synthesis performance.
Hire Me 😊

- I am applying for PhD positions for **Fall 2021**!

- a.yang@queensu.ca
- https://aidanby.github.io/
- @AidanYang5  🦅