Open Vocabulary Learning on Source Code with a Graph-Structured Cache

Milan Cvitkovic  
Caltech, Amazon Web Services

Badal Singh  
Amazon Web Services

Anima Anandkumar  
Caltech

ICML, 2019-6-12
Open Vocabulary Learning

**Goal:** Models that can reason over flexible sets of inputs and outputs

Standard, closed vocabulary model

1 of 400k word embeddings $\rightarrow$ 1 of 400k words

Open vocabulary

Any words $\rightarrow$ Any words
Open Vocabulary Learning

Motivation: Tasks on source code
Example: Variable naming

Needs an open vocabulary
In our data, 28% of variable names contain out–of–vocabulary word
Graph-Structured Cache

**Strategy:** Represent distinct words and usages with graph structure, process with GNN

```python
def get_jupyter_addr():
    jupyter_addr = 'localhost' if is_serving() else None
    return jupyter_addr
```

Same input, represented using a Graph-Structured Cache
Full Model for Tasks on Source Code

Strategy from recent work [1]

Parse code into AST

Augment AST with semantic information

Full Model for Tasks on Source Code

Our main contribution to prior work
Full Model for Tasks on Source Code

Input:
/** SomeFile.java
public void addFoo(Foo foo){
    this.myBaz.add(foo);
}

Parse code into AST
Augment AST with semantic information
Add Graph-Structured Cache
Convert all nodes to vectors, process with GNN

Output (Depends on task)
### Experiment: Variable Naming Task

- Full-name reproduction accuracy (and top 5 accuracy):

<table>
<thead>
<tr>
<th></th>
<th>Closed Vocab</th>
<th>CharCNN</th>
<th>Pointer Sentinel</th>
<th>GSC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Seen repos</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AST</td>
<td>0.23 (0.31)</td>
<td>0.22 (0.28)</td>
<td>0.19 (0.33)</td>
<td>0.49 (0.67)</td>
</tr>
<tr>
<td>AugAST</td>
<td>0.19 (0.26)</td>
<td>0.20 (0.27)</td>
<td>0.26 (0.40)</td>
<td><strong>0.53 (0.69)</strong></td>
</tr>
<tr>
<td><strong>Unseen repos</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AST</td>
<td>0.05 (0.07)</td>
<td>0.06 (0.09)</td>
<td>0.06 (0.11)</td>
<td>0.38 (0.53)</td>
</tr>
<tr>
<td>AugAST</td>
<td>0.04 (0.07)</td>
<td>0.06 (0.08)</td>
<td>0.08 (0.14)</td>
<td><strong>0.41 (0.57)</strong></td>
</tr>
</tbody>
</table>

For other tasks and experiments, see our poster or paper.
Takeaways

Graph-Structured Caches are an appealing strategy for open vocabulary learning

- Whatever your current embedding strategy, GSC + GNN can augment it
- No free lunch! About 30% training slowdown.
- But helps in all cases we tried, sometimes significantly
Acknowledgments

- Badal Singh, Anima Anandkumar
- Miltos Allamanis
- Hyokun Yun
- Haibin Lin

Our code, for use on your code
