

# Auto Comments: Generating Java code comments

R. Navin\*, J. Katzy\*, R. Skoulos\*, T. Pfann\*

\*Delft Institute of Computer Science, TU Delft

## Motivation & Goal

- In software development and maintenance, developers spend around 59% of their time on program comprehension activities.
- Automatically generate human readable comments for code snippets.
- With DeepCom as baseline, we propose,
  - Method-1: Replication of code2seq, with added capability to generate natural languages as comments.
  - Method-2: Learn on modified ASTs, solving Out-of-Vocabulary problems.

## Experiment Setup

Java methods are parsed into ASTs, which are encoded and passed to a Encoder-Decoder sequence to sequence neural network based upon bidirectional LSTMS (a code2seq based architecture).

```
1 public static void add(int VAR0, int VAR1) {  
2     return VAR0 + VAR1;  
3 }
```

Listing 1: Java example

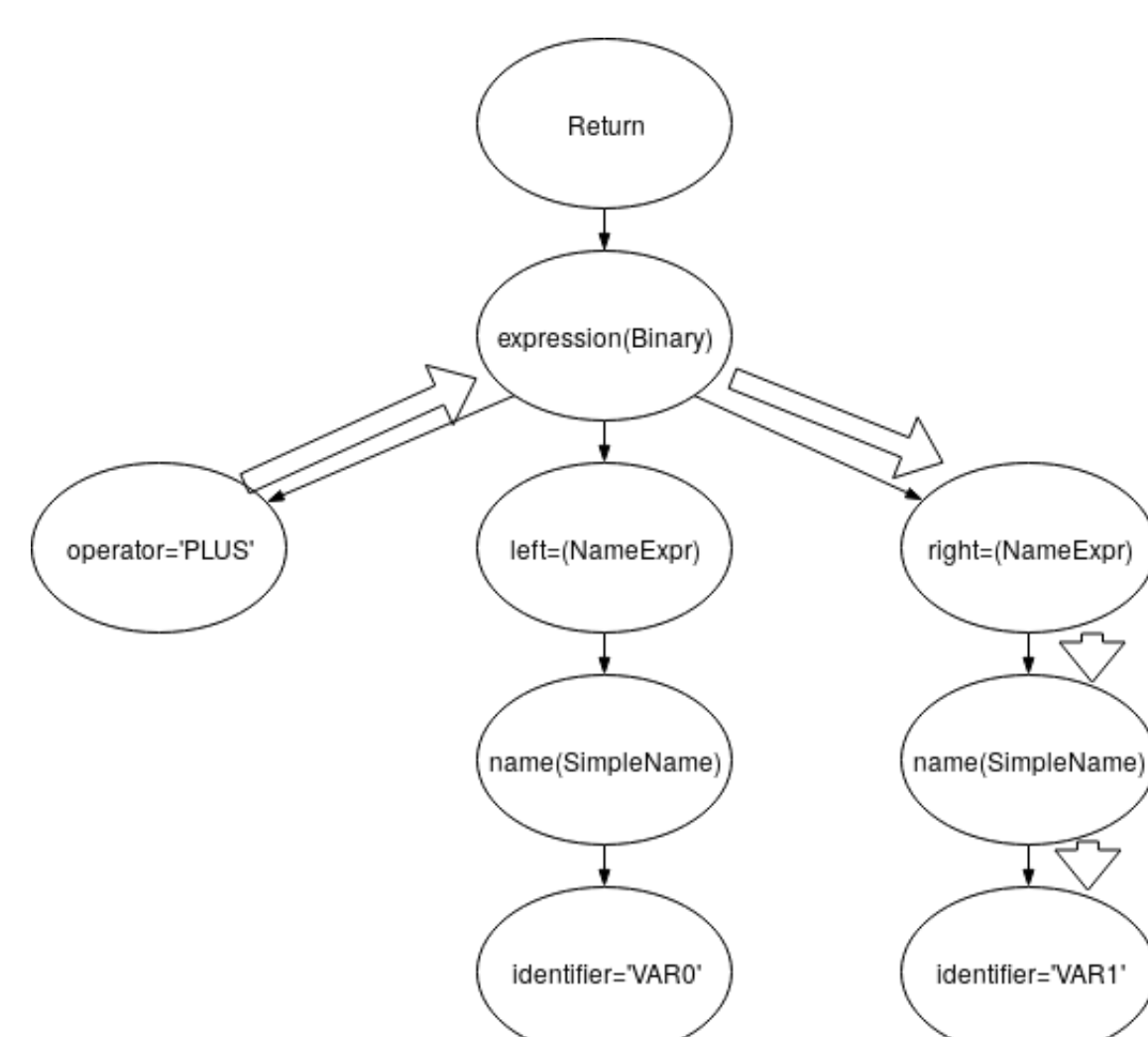


Figure 1: Example AST of Function, the example path has been superimposed with thick arrows.

### Dataset

# Methods	# All tokens	# All identifiers	#Unique tokens	#Unique identifiers
588,108	44,378,497	13,779,297	794,711	794,621

Table 1: Statistics for code-snippets in DeepComm dataset

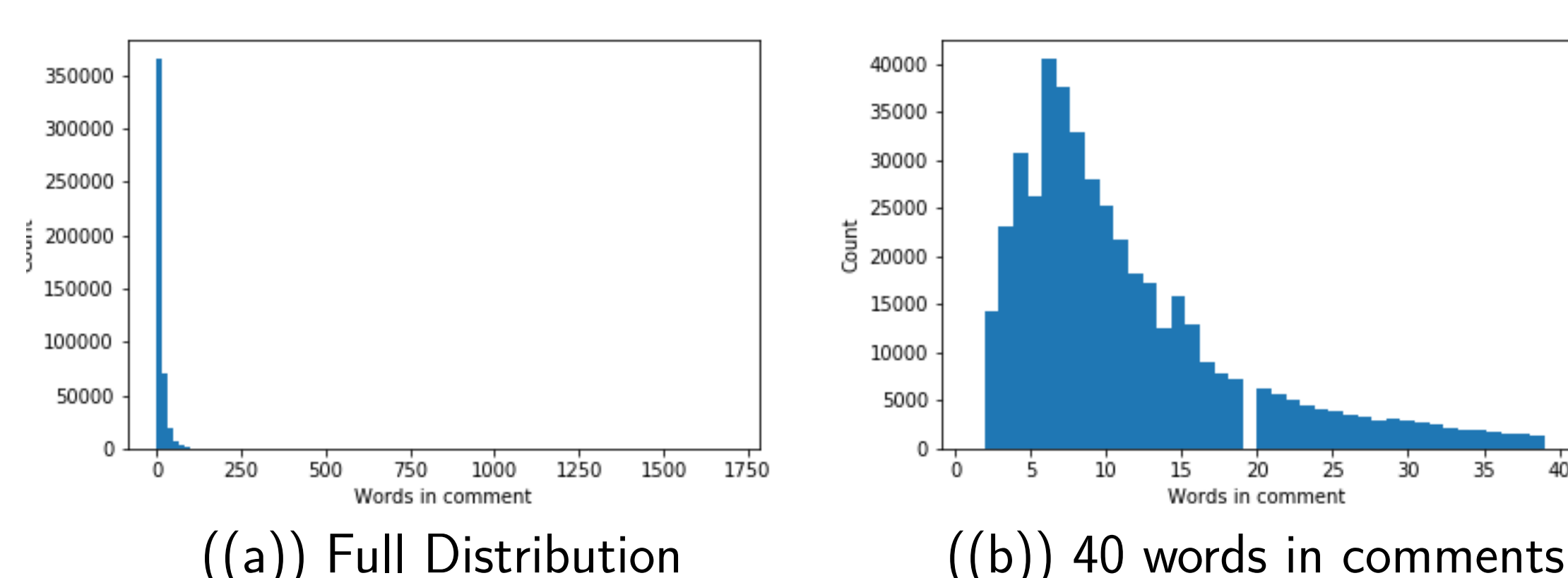


Figure 2: Dataset distribution of target comment lengths.

### Training

- Setup
  - Learning rate 0.01 with 0.05 decay every epoch.

- Embeddings size: 128, Encoder size: 256, Decoder size: 640, Batch size: 128.
- Trained for 100 epochs. Early stopping if no improvement for 10 epochs.

- Method - 1: Code2Seq model with comments as target sequence.
- Method - 2: Same as method 1 but with variable names in ASTs.
- Evaluation: BLEU-4 score

## Results

Approaches BLEU-4 score

DeepCom	38.17
Method-1	6.08
Method-2	10.02

Table 2: Evaluation results on Java Methods

Code	Comments
<pre>public static void sort(Comparable[] a) {     int n=a.length;     for (int i=1; i &lt; n; i++) {         Comparable v=a[i];         int lo=0, hi=i;         while (lo &lt; hi) { ... }         v=a[hi];         assert isSorted(a);     } }</pre>	<p><b>Hand-Written:</b> Rearranges the array in ascending order, using the natural order.</p> <p><b>DeepCom:</b> Sorts the array in ascending order, using the natural order.</p> <p><b>Method-1:</b> Sorts the var with the given.</p> <p><b>Method-2:</b> Sort a set of entries based.</p>
<pre>public boolean isEmpty() {     return root == null; }</pre>	<p><b>Hand-Written:</b> Is this symbol table empty?</p> <p><b>DeepCom:</b> Returns true if the symbol is empty.</p> <p><b>Method-1:</b> Returns true if the tree is.</p> <p><b>Method-2:</b> Returns true if the symbol empty.</p>
<pre>public void unlisten(String pattern) {     @SuppressWarnings("unchecked")     List&lt;Listener&gt; listeners=listeners.get(pattern);     if(listeners==null) {         listeners.remove(pattern);     }     Client.onError(Topic.RECORD, Event.NOT_LISTENING, pattern); }</pre>	<p><b>Hand-Written:</b> Removes a listener that was previously registered with listenFor-Subscriptions.</p> <p><b>DeepCom:</b> It can be called when the product only or refresh has ended.</p> <p><b>Method-1:</b> Removes a var from the topic.</p> <p><b>Method-2:</b> Removes the message from the listener.</p>
<pre>protected void createItemLayout() {     if (mItemLayout == null) {         mItemLayout=new LinearLayout(getContext());         mItemLayout.setOrientation(LinearLayout.VERTICAL);     } }</pre>	<p><b>Hand-Written:</b> Creates item layout if necessary.</p> <p><b>DeepCom:</b> Creates item layouts if any parameters.</p> <p><b>Method-1:</b> Creates the layouts layout.</p> <p><b>Method-2:</b> Creates item layouts if necessary.</p>

Figure 3: Comments Generated by models.

## Discussion

- Probable reasons for poor BLEU score [Table-2],
  - Imbalanced distribution 2 of target comment lengths in the dataset.
  - Code2Seq architecture - Built to predict function names.
- Performance of Method - 2, proves to be good solution to Out-of-Vocabulary problems.
- Model learnt the syntactic and semantic concepts from codes. [Fig - 3]
- But, Incapable of generating longer comments (>6 words).

## Conclusion

- Contributions: code2seq based AutoComments, and, AST extraction to solve Out-of-Vocabulary.
- Future Research,
  - Balanced dataset - w.r.t. target comment lengths.
  - More experiments with decoder, for generating better comments from the learnt code semantics and syntaxes.

