

# AutoComments:

## *Comment Generation in Java Code*

### **Group 3**

Navin Raj Prabhu,  
Jonathan Katzy,  
Rafail Skoulos,  
Thomas Pfann

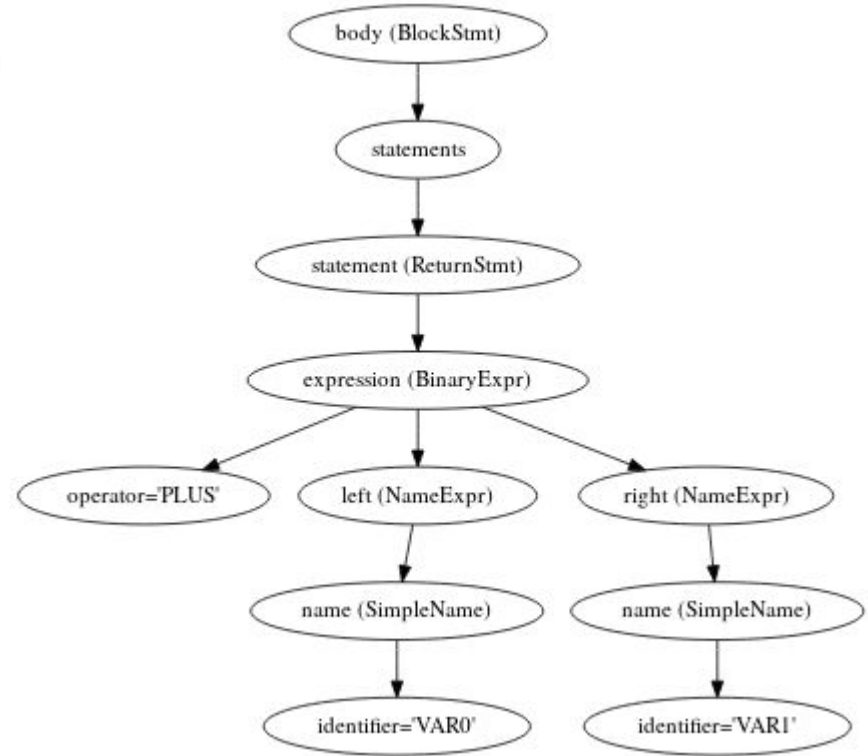
# Motivation & Goal

- In software development and maintenance, developers spend around 59% of their time on program comprehension activities.
- A comment generator model based on code2seq[1] for comment generation in Java code.
- With DeepCom[2] 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[3] problems.

1. Uri Alon, Shaked Brody, Omer Levy, and Eran Yahav. code2seq: Generating sequences from structured representations of code.
2. Xing Hu, Ge Li, Xin Xia, David Lo, and Zhi Jin. Deepcode comment generation. In Proceedings of the 26th Conference on Program Comprehension, ACM, 2018.
3. Yu Zhou, Xin Yan, Wenhua Yang, Taolue Chen, and Zhiqiu Huang. Augmenting java method comments generation with context information based on neural networks. Journal of Systems and Software, 2019.

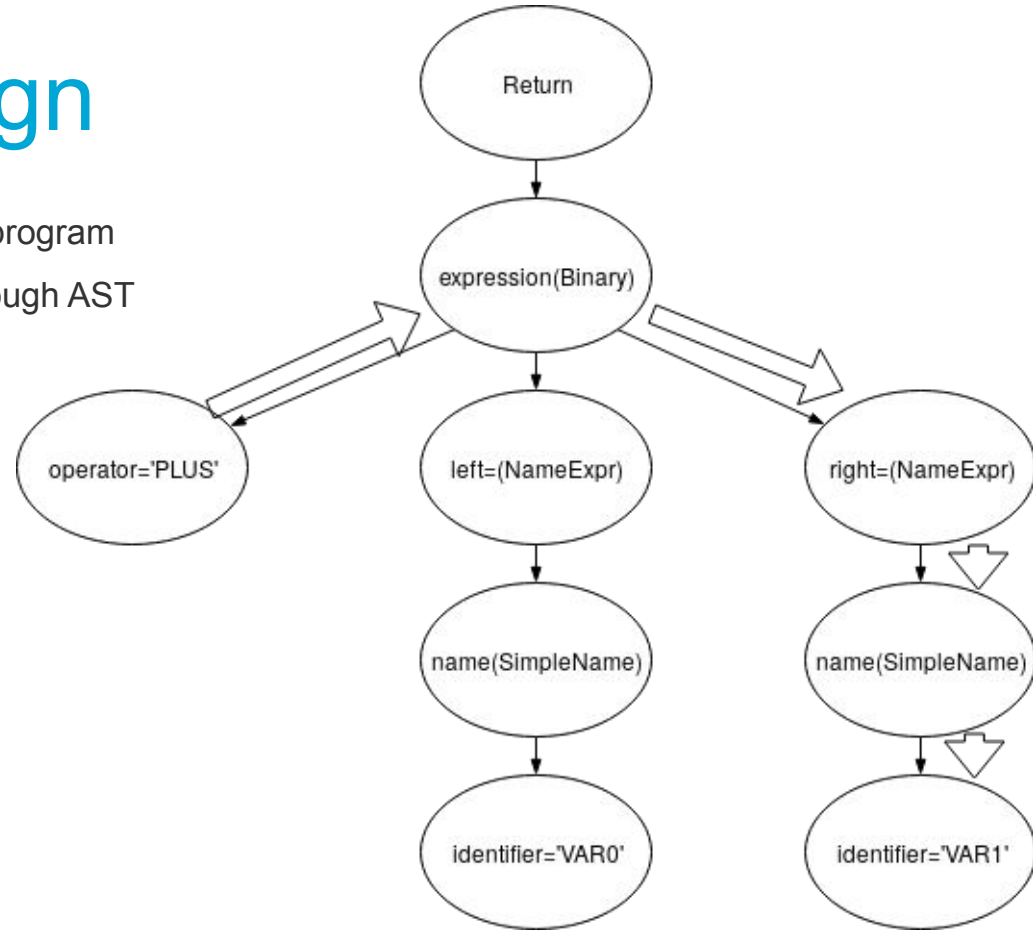
# Model Design

- Generate AST from java program



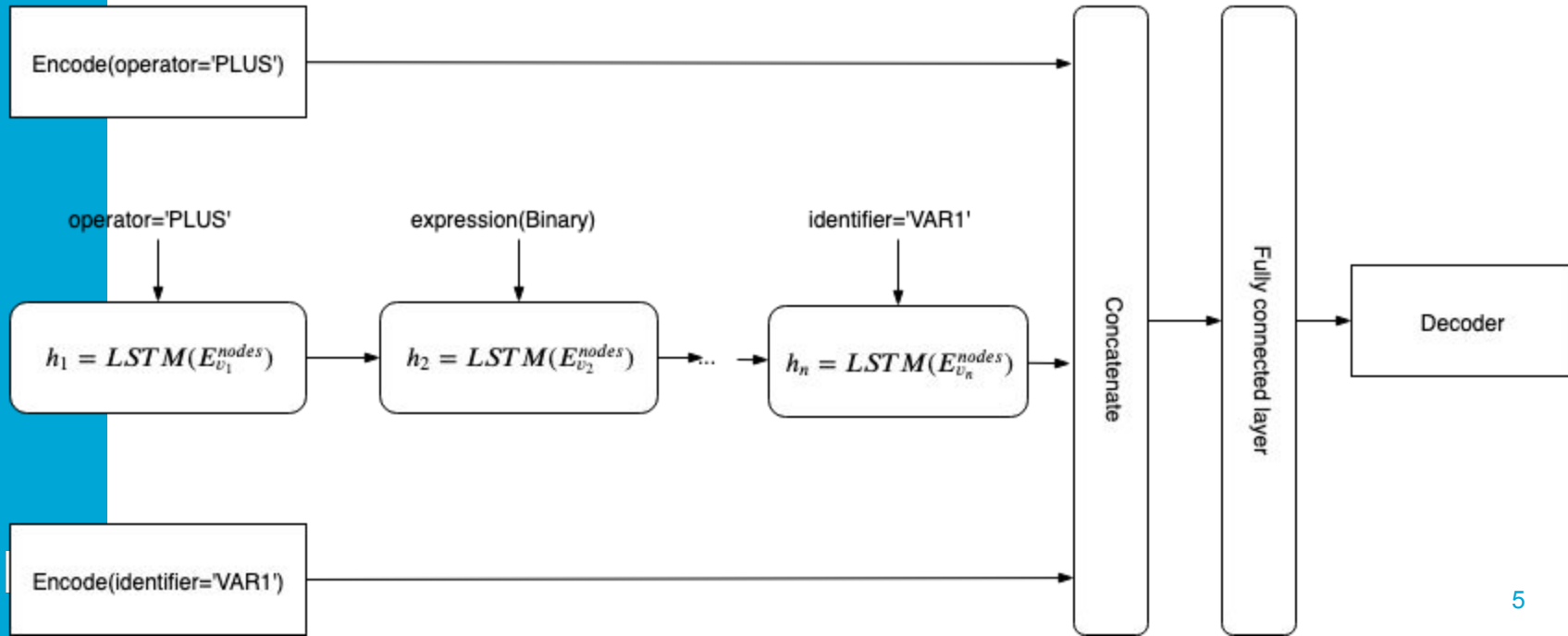
# Model Design

- Generate AST from java program
- Create random walks through AST



# Model Design

- Embed these random paths, And then concatenate the first node of the path, last node of the path and the path itself.



# Experiment Setup

- **Setup**
  - Cross-entropy loss with a Nesterov momentum of 0.95.
  - Learning rate 0.01 with 0.05 decay every epoch.
  - Embeddings size: 128, Encoder size: 256, Decoder size: 640, Batch size: 128.
  - Applied dropout of 0.3 and a recurrent dropout of 0.5 on the LSTM encoding the AST paths.
  - 200 paths were chosen from each of the extracted ASTs
  - 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

- Our methods failed to out-perform the baseline DeepCom model (In terms of BLEU -4). [Table -1]
- At the same time, Method - 2 improves over Method - 1. [Table -2]
- While quantitative results reveal the overall performances of models, inspection of actual comments generated (by methods) may be more informative.

Approaches	BLEU-4 score
DeepCom	38.17
Method-1	6.08
Method-2	10.02

Table - 1 : Evaluations Results

Approaches	Precision	Recall	F1
Method-1	36.26	21.56	27.04
Method-2	46.94	27.44	34.63

Table - 2 : Model Performance

# Results

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){ ... }         ...     }     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 boolean contains(int key){     return rank(key) != -1; } </pre>	<p><b>Hand-Written:</b> Is the key in this set of integers?</p> <p><b>DeepCom:</b> Checks whether the given object is contained within the given set.</p> <p><b>Method-1:</b> Returns true if the key is.</p> <p><b>Method-2 :</b> Returns true if this symbol contains.</p>
<pre> public void unlisten(String pattern){     UtilListener listener=listeners.get(pattern);     if(listener!=null){         listener.destroy();         listeners.remove(pattern);     }else{         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 createItemsLayout(){     if (mItemsLayout == null){         mItemsLayout=new LinearLayout(getContext());         mItemsLayout.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>

Table - 2 : Comments Generated by Models



# Discussion

- Probable reasons for poor BLEU score [Table-1],
  - Imbalanced distribution of target comment lengths in the dataset.
  - Code2Seq architecture - Built to predict function names.

<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> <u>Creates the layouts layout.</u></p> <p><b>Method-2:</b> <u>Creates item layouts if necessary.</u></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> <u>Returns true if the tree is.</u></p> <p><b>Method-2:</b> <u>Returns true if the symbol empty.</u></p>

Figure - 1 : Comment Generated

- Nevertheless, From Table - 3,
  - Performance of Method - 2, proves to be good solution to *Out-of-Vocabulary* problems.
  - Model has learnt the *syntactic and semantic* meanings from the code. Example: Fig - 1.
  - Incapable of generating longer comments (>6 words).

# Conclusion

- Contribution of our research,
  - Code2Seq based Comment Generation - AutoComments
  - AST extraction to solve *Out-of-Vocabulary*.
- Drawbacks,
  - Poor BLEU-4 scores on evaluation.
  - Model incapable of generating comments >6 words.
- Future Research,
  - Balanced dataset - w.r.t. target comment lengths.
  - Better encoding technique on target comments.
  - More experiments with decoder, for generating better comments from the learnt code semantics and syntaxes.