HierarchyNet: Learning to Summarize Source Code with Heterogeneous Representations

Minh Huynh Nguyen [1], Nghi D. Q. Bui [1,2], Truong Son Hy [1,3],
Long Tran Thanh [4], Tien N. Nguyen [5]

[1] FPT Software AI Center; [2] Fulbright University, Viet Nam; [3] Indiana State University, USA;
[4] University of Warwick, UK; [5] The University of Texas at Dallas, USA

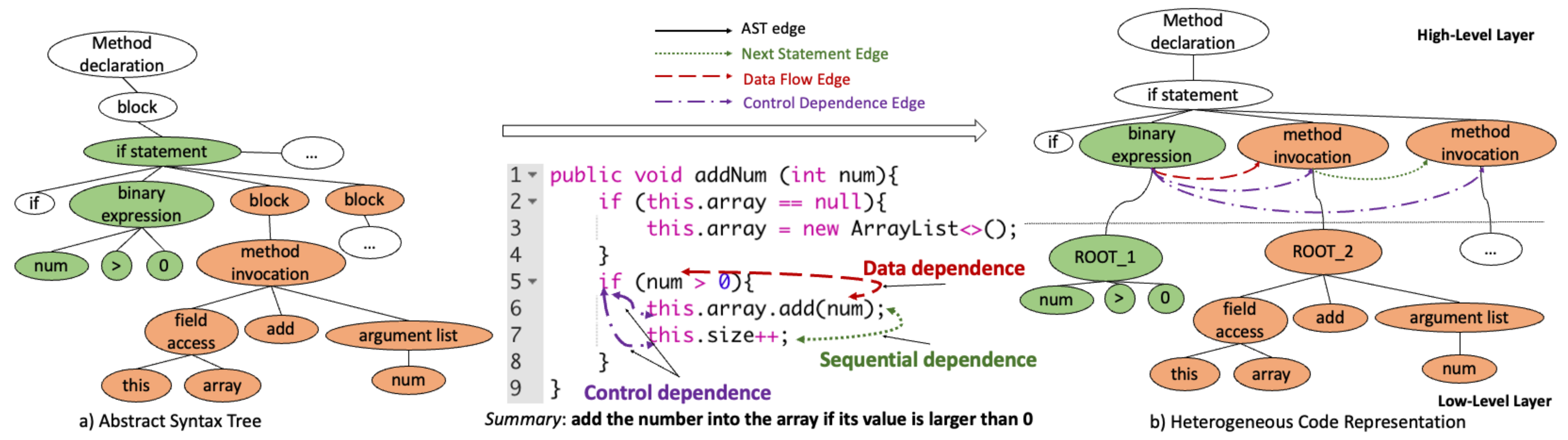
Contributions

In this paper, we propose a novel code summarization approach utilizing:

- **Heterogeneous Code Representations (HCRs)** adeptly capturing essential code features at lexical, syntactic, and semantic levels within a hierarchical structure
- **HierarchyNet** processing each layer of the HCR separately, employing a Heterogeneous Graph Transformer, a Tree-based CNN, and a Transformer Encoder.

Motivation

- Existing code summarization approaches often overlook the critical consideration of the interplay of dependencies among code elements and code hierarchy.
- Effective summarization necessitates a holistic analysis of code snippets from three distinct aspects: lexical, syntactic, and semantic information.



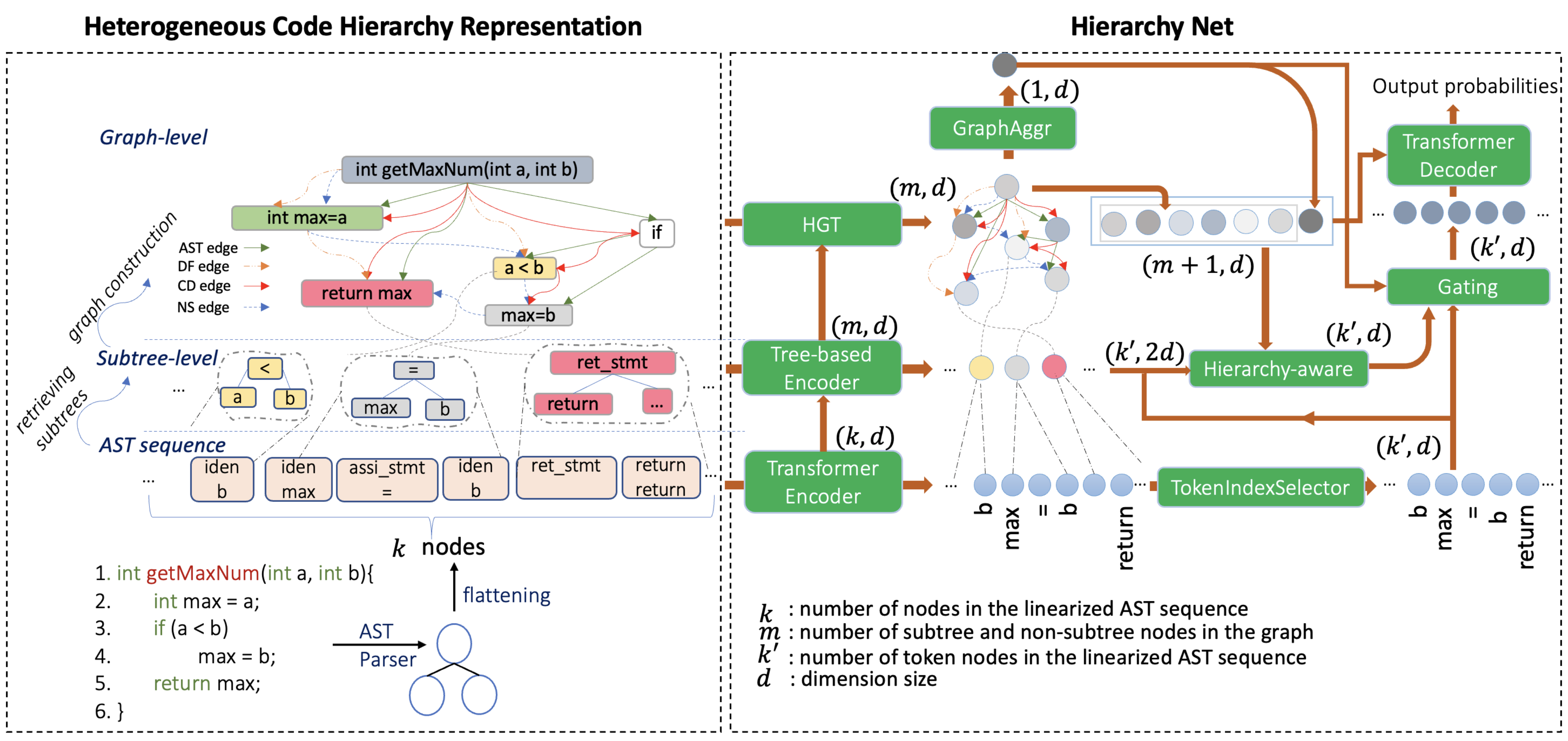
```
public void addNum (int num){
    if (this.array == null){
        this.array = new ArrayList<>();
    }
    if (num > 0){
        this.array.add(num);
        this.size++;
    }
}
```

a) Abstract Syntax Tree

Summary: add the number into the array if its value is larger than 0

b) Heterogeneous Code Representation

Method



```
int getMaxNum(int a, int b){
    int max = a;
    if (a < b)
        max = b;
    return max;
}
```

k : number of nodes in the linearized AST sequence
m : number of subtree and non-subtree nodes in the graph
k' : number of token nodes in the linearized AST sequence
d : dimension size

Heterogeneous Code Representation The first layer, ”Linearized AST Sequence,” comprises serialized AST nodes. The second, ”Subtree-level,” represents statement and expression-level elements. Lastly, the ”Graph level” represents a high-level graph with semantic edges like dependencies.

HierarchyNet HCR utilizes neural networks for each layer. Information is aggregated across layers using a Hierarchy-aware cross-attention layer, with a gating layer to balance lexical and hierarchical context before input to a Transformer Decoder.

Conclusion

We introduce an innovative framework for code summarization, combining HCRs and HierarchyNet. HCRs inherently capture key features of source code from lexical, syntactic, and semantic meanings, while HierarchyNet is tailored to processing HCRs. For future work, we aim to investigate:

- Provide an analysis of the explainability
- Evaluate on other code-related tasks

Human Evaluation

To consolidate the effectiveness of our method, we carry out a user study, utilizing a linear 3-point rating scale.

Methods	Naturalness	Usefulness
CAST	2.76	2.48
PA-former	2.77	2.50
HierarchyNet	**2.81**	**2.52**

Automated Evaluation

We evaluate on four datasets: TL-CodeSum, DeepCom, FunCom-50, and FunCom. Due to the space limit, we just show evaluation results on DeepCom and FunCom-50.

Model	DeepCom			FunCom-50		
	BLEU	Meteor	Rouge-L	BLEU	Meteor	Rouge-L
Training from scratch						
NCS	37.13	25.05	54.80	43.36	27.54	60.41
TPTrans	37.25	25.02	55.00	43.45	27.61	60.57
CAST	38.03	25.27	54.95	43.58	27.67	60.52
PA-former	39.67	26.21	56.18	44.65	28.27	61.45
Fine-tuning						
CodeBERT-base	37.42	25.49	55.07	46.20	30.51	61.43
CodeT5-base	38.60	26.30	56.31	46.88	30.72	61.47
In-context Learning						
CodeGen-Multi 2B (two-shot)	17.81	13.81	24.62	21.78	14.78	26.89
StarCoder (two-shot)	19.29	16.07	28.09	25.18	18.45	32.59
CodeLlama 13B (two-shot)	20.29	16.14	39.63	21.52	16.52	36.49
HierarchyNet	**43.64**	**29.22**	**59.00**	**51.12**	**34.13**	**65.43**

Analysis

Study on HierarchyNet We aim to demonstrate the significance of our proposed layers in HierarchyNet on the TL-CodeSum dataset.

Method	BLEU	Meteor	Rouge-L	Cider
HierarchyNet	**48.01**	**30.30**	**57.90**	**4.20**
w/o Hierarchy-aware	46.63	29.49	56.63	4.03
w/o TokenIndexSelector	45.70	28.39	55.06	3.93

Comparisons with LLMs Given that LLMs may potentially generate responses longer and more detailed than the ground truth, we aim to demonstrate the fairness of our evaluation.

Model	Average word count
StarCoder (zero-shot)	10.64
StarCoder (one-shot)	7.59
CodeGen 2B (zero-shot)	4.95
CodeGen 2B (one-shot)	8.46
References	9.97

Qualitative Example

A code snippet sample.

```
@Override public void start (Stage stage) throws Exception {
    CategoryDataset dataset = createDataset();
    JFreeChart chart = createChart(dataset);
    ChartViewer viewer = new ChartViewer(chart);
    viewer.addChartMouseListener(this);
    stage.setScene(new Scene(viewer));
    stage.setTitle(<str>);
    stage.setWidth(700);
    stage.setHeight(390);
    stage.show();
}
```

Summaries from several variants of HCRs.

ID	Options	Sentence
1	Tokens	creates a chart bar chart (clicked)
2	Tokens + Subtrees	creates and displays a chart viewer
3	Tokens + Subtrees + Graph (only AST edges)	adds a chart viewer to the stage
4	Tokens + Subtrees + Graph (full of edge types) (ours)	adds a chart viewer to the stage and displays it
	Ground-truth	adds a chart viewer to the stage and displays it

Source code

We release our implementation at `https://github.com/FSoft-AI4Code/HierarchyNet`.