



# ASTCODA: Abstract Syntax Tree Convolutions Operating on Domain Attention

Liudmila Paskonova <sup>(1)</sup>, Lomonosov Moscow State University

Supervised by Dr. Alexander Chernov <sup>(2)</sup>

## Background: Approaches to code segmentation have limitations

- require fixed segment annotations
- miss domain nuances
- limited to specific programming languages

## Problem:

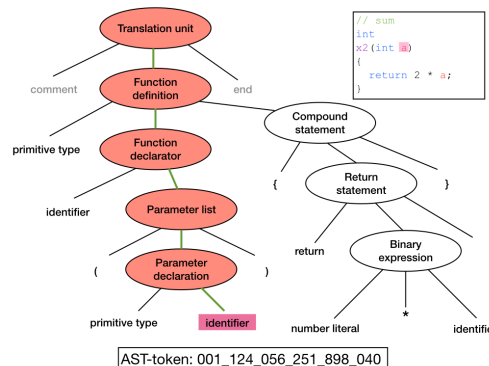
We want to extract features from *unlabeled* source code *automatically*.

## Solution:

Features are sequences of consecutive tokens. We employ AST-based tokenizer and a CNN with attention to assess their importance. Our framework is **AST-based, self-contained, domain-aware** and **multi-language**.

## AST-based tokenization preserves the structure of the program code

- To construct AST we use Tree-Sitter
- AST-tokenization worked better than word-based approaches

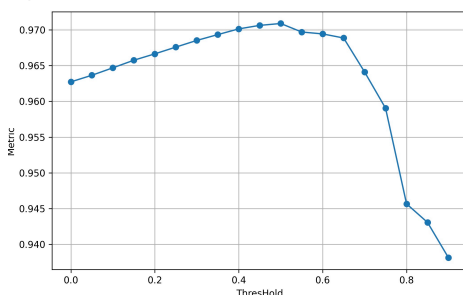


## Higher attention weights highlight more important features

$$\text{RSDM}(\text{pred}, \text{gt}) = 1 - \frac{\text{distance}(\text{pred}, \text{gt})}{\text{worst\_distance}(L, \text{gt})}$$

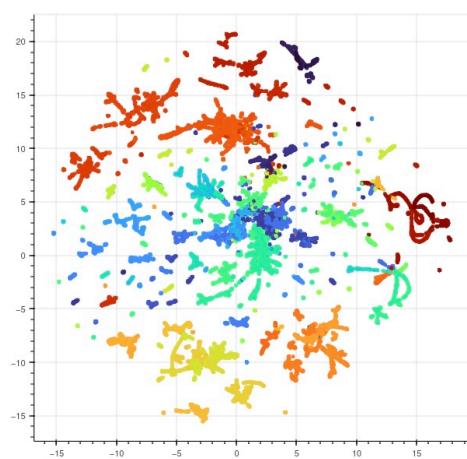
$$\text{distance}(\text{pred}, \text{gt}) = \sum_{i=0}^{M-1} \min_{j \in \{0 \dots K-1\}} |\text{pred}_i - \text{gt}_j|$$

$\tilde{\alpha}_i \geq \text{threshold} \implies$  feature  $i$  is more likely to correspond the predicted class



## Embeddings of neighboring AST-tokens form syntax clusters

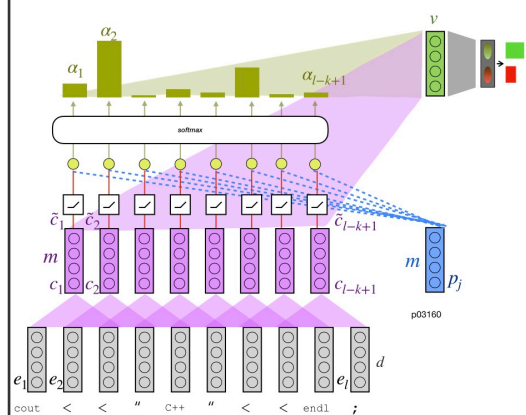
This helps solve OOV problem.



## Our model finds logical errors

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int main()
4 {
5     ios_base::sync_with_stdio(0);
6     cin.tie(0);cout.tie(0);
7     long i=0;
8     long n;
9     vector<long> r;
10    cin>>n;
11    long J;
12    for (int j=0;j<n;j++)
13    {
14        cin>>J;r.push_back(J);
15    }
16    J=0;
17    while(i<n)
18    {
19        if (abs(r.at(i)-r.at(i+1))>abs(r.at(i+2)-r.at(i)))
20        {
21            i+=2;
22            J+=abs(r.at(i+2)-r.at(i));
23        }
24        else
25        {
26            J+=abs(r.at(i+1)-r.at(i));
27        }
28    }
29    cout<<J<<endl;
30    return 0;
31 }
```

## Model Architecture



## Code, data and pretrained models



<https://github.com/Liudmila-Paskonova/ASTCODA>

Task	Source	Samples	Domain	#	Class	#	lang.
Vulnerability detection	FormAI dataset	336523	Entire dataset	1	Correct/ Vulnerable	2	C
Error localization in student programming submissions	Project CodeNet	449950	Problem	30	Correct/ Partial Solution	2	C++