AUTOMATIC REPAIR AND TYPE BINDING OF UNDECLAREDVARIABLES USING NEURAL NETWORKS

Hung PHAN
Venkatesh Mohan

Input and Output

Input

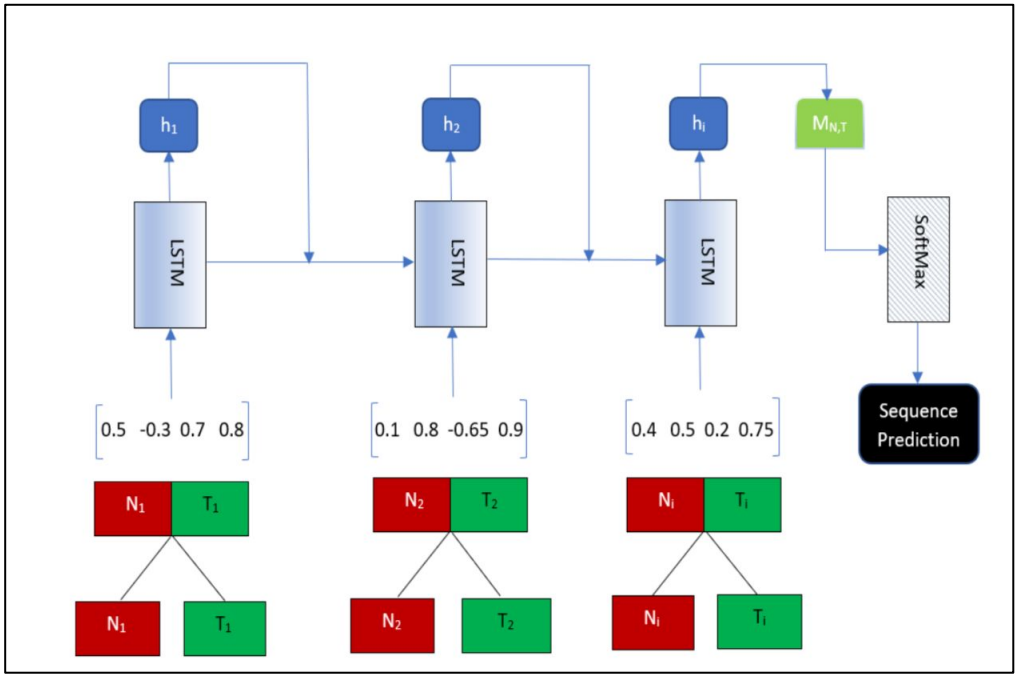
```
1  int main()
2  {
3  int n[1000],a[500],nm,i,j,ln,
       flag=0;
4  scanf("%d\n",&ln);
5  scanf("%d\n",&nm);
6  for(i=0;i<500;i++)
7  {
8       a[i]=0;
9  }
10 for(i=0;i<nm;i++)
11 {
12      scanf("%d ",&nm);
13      c=n[i];
14 }
15 return 0;
16 }
```

LSTM-RNN

Output

```
1  int main()
2  {
3  int c;
4  int n[1000];
5  int a[500];
6  int nm;
7  int i;
8  int j;
9  int ln;
10 int flag = 0;
11 scanf("%d""\n", &ln);
12 scanf("%d""\n", &nm);
13 for (i = 0; i < 500; i++)
14 {
15      a[i] = 0;
16 }
17 for (i = 0; i < nm; i++)
18 {
19      scanf("%d" , &nm);
20      c = n[i];
21 }
22 return 0;
23 }
```

How LSTM-RNN works



Evaluation

	Identified	Not Identified	Correctly Identified (True Positive)	Wrongly Identified (False Positive)	Correctly Identified + Correct Type Inferred (Fixed)	Wrongly Identified + Wrong Type Inferred (Not Fixed)	Total
Undeclared Variables and Arrays	887(83.7%)	172	857(80.9%)	202	844(79.7%)	215	1059
Undeclared variables - Main function	N/A	N/A	566(99.1%)	5	560(98%)	11	571
Undeclared variables - Multiple functions	N/A	N/A	179(91.7%)	16	172(88.2%)	23	195
Undeclared Arrays - Main functions	N/A	N/A	90(96.8%)	3	90(96.8%)	3	93
Undeclared Arrays - Multiple functions	N/A	N/A	22(78.5%)	6	22(78.5%)	6	28

Table 1: Analysis results of both the undeclared variables and arrays

Ongoing works

We are extending the work for types of bugs which cannot be detected by the compilation:

- Uninitialized variables.
- Incorrect arguments used in Method Call.