http://lamda.nju.edu.cn

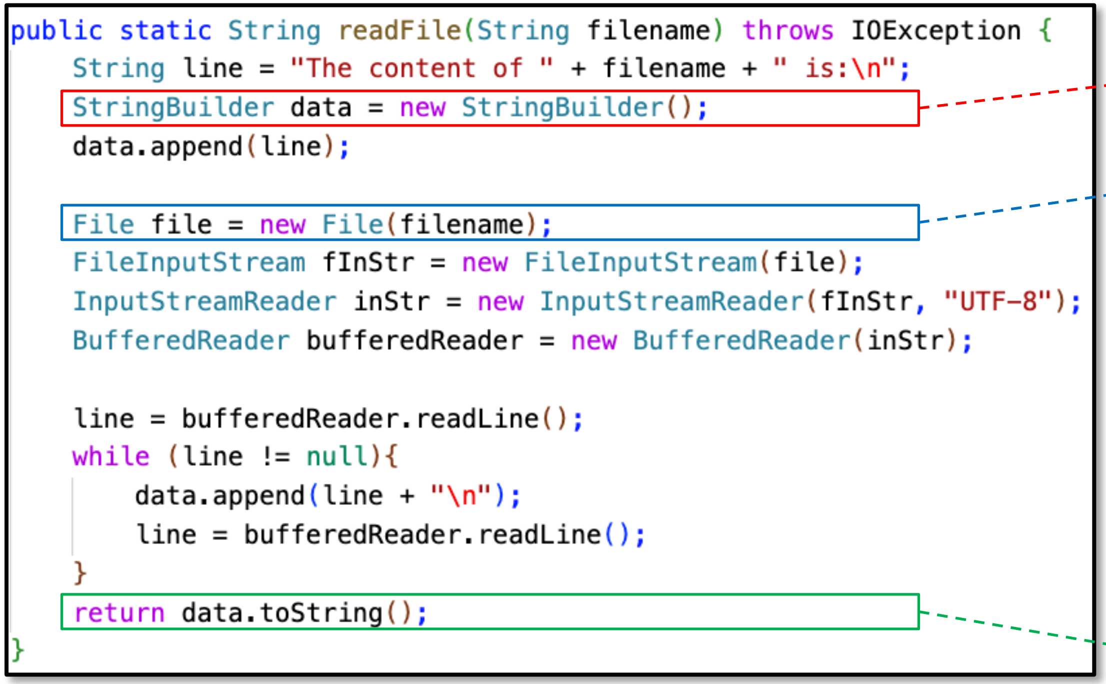
Capturing the Long-Distance Dependency in the Control Flow Graph via Structural-Guided Attention for Bug Localization

Yi-Fan Ma, **Yali Du**, and **Ming Li**
LAMDA Group
National Key Laboratory for Novel Software Technology,
Nanjing University
{mayf, duyl, lim}@lamda.nju.edu.cn

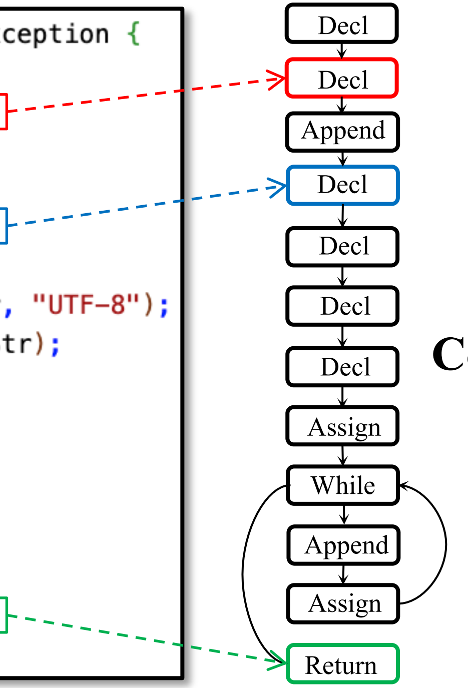
核心目标 – 建模代码的控制流以表征程序功能

LAMDA Learning And Mining from DatA
http://lamda.nju.edu.cn

控制流结构表达了**程序的执行逻辑**，本质上是**程序算法的语句层级表示**



```
public static String readFile(String filename) throws IOException {
    String line = "The content of " + filename + " is:\n";
    StringBuilder data = new StringBuilder();
    data.append(line);

    File file = new File(filename);
    FileInputStream fInStr = new FileInputStream(file);
    InputStreamReader inStr = new InputStreamReader(fInStr, "UTF-8");
    BufferedReader bufferedReader = new BufferedReader(inStr);

    line = bufferedReader.readLine();
    while (line != null){
        data.append(line + "\n");
        line = bufferedReader.readLine();
    }
    return data.toString();
}
```

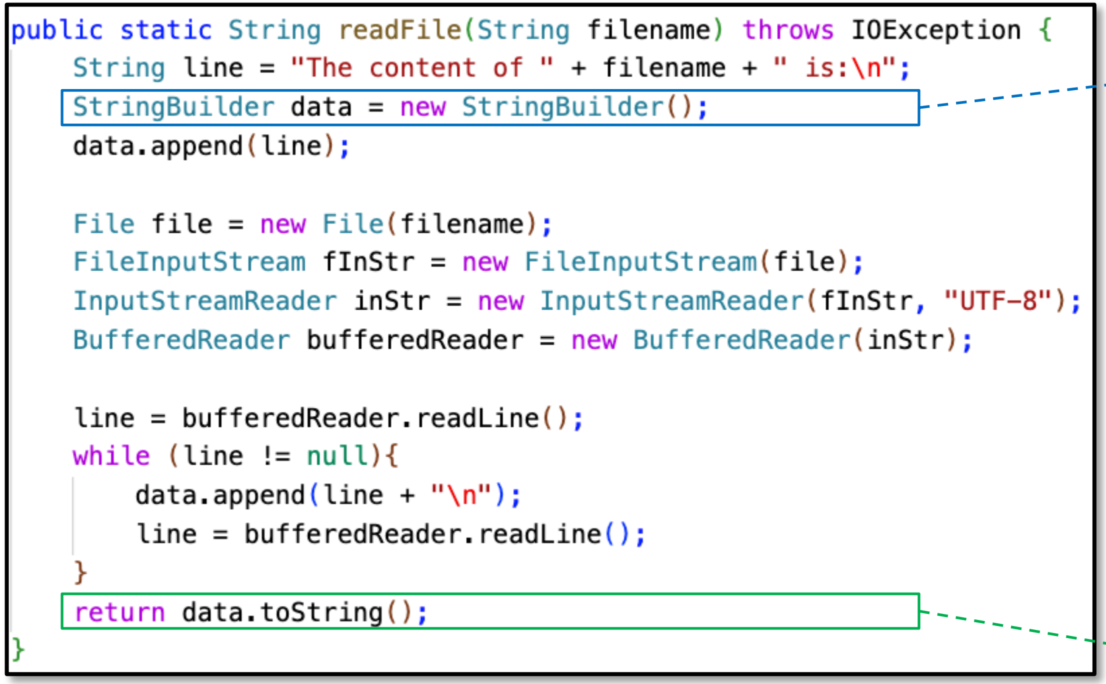


代码控制流图
Control Flow Graph,
CFG
(Allen, 1970)

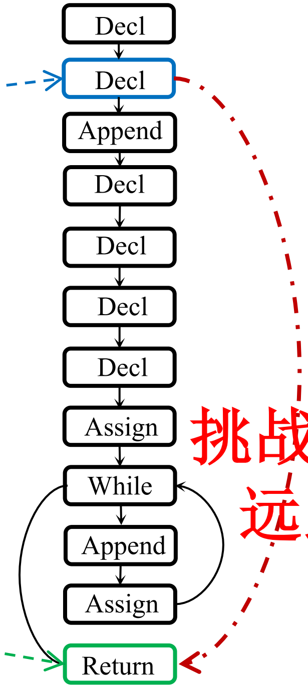
现状与挑战1

Learning And Mining from DatA
http://lamda.nju.edu.cn

- **现状1：** 语句之间存在**远距离依赖关系(long-distance dependency)**



```java
public static String readFile(String filename) throws IOException {
    String line = "The content of " + filename + " is:\n";
    StringBuilder data = new StringBuilder();
    data.append(line);

    File file = new File(filename);
    FileInputStream fInStr = new FileInputStream(file);
    InputStreamReader inStr = new InputStreamReader(fInStr, "UTF-8");
    BufferedReader bufferedReader = new BufferedReader(inStr);

    line = bufferedReader.readLine();
    while (line != null){
        data.append(line + "\n");
        line = bufferedReader.readLine();
    }
    return data.toString();
}
```



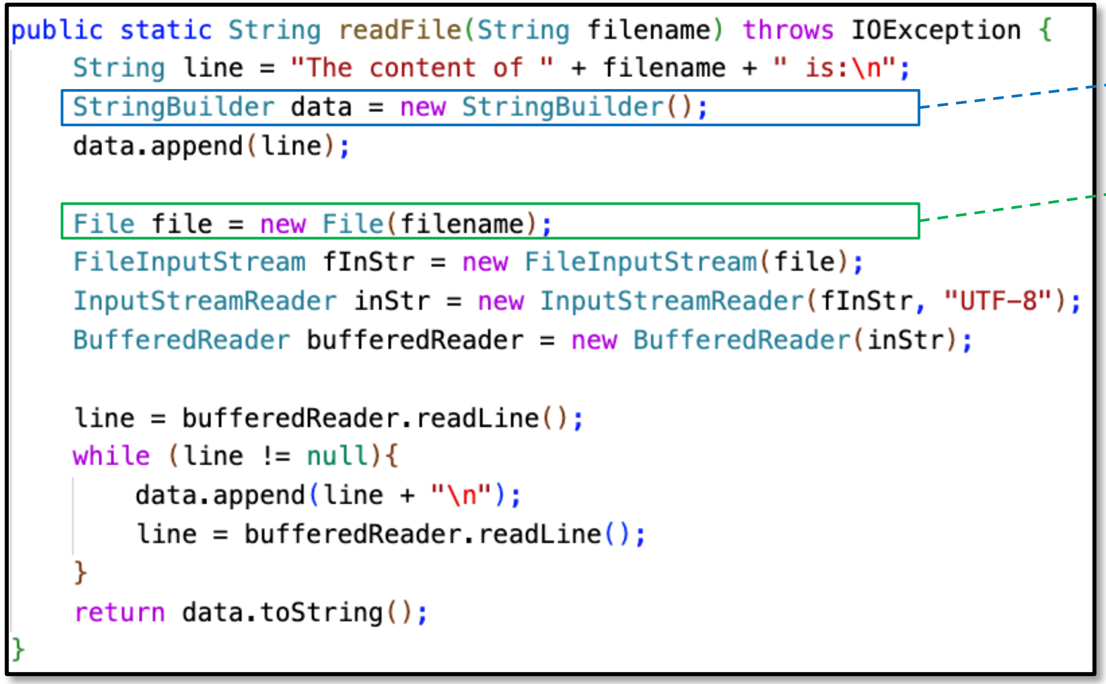
逐步传递过程中因为距离长，导致信息易丢失

挑战1：如何捕获远距离依赖？

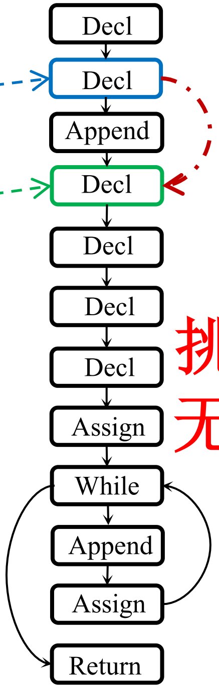
现状与挑战2

Learning And Mining from DatA
http://lamda.nju.edu.cn

- **现状2：** 语句之间存在**互相独立**的情况



```java
public static String readFile(String filename) throws IOException {
    String line = "The content of " + filename + " is:\n";
    StringBuilder data = new StringBuilder();
    data.append(line);

    File file = new File(filename);
    FileInputStream fInStr = new FileInputStream(file);
    InputStreamReader inStr = new InputStreamReader(fInStr, "UTF-8");
    BufferedReader bufferedReader = new BufferedReader(inStr);

    line = bufferedReader.readLine();
    while (line != null){
        data.append(line + "\n");
        line = bufferedReader.readLine();
    }
    return data.toString();
}
```

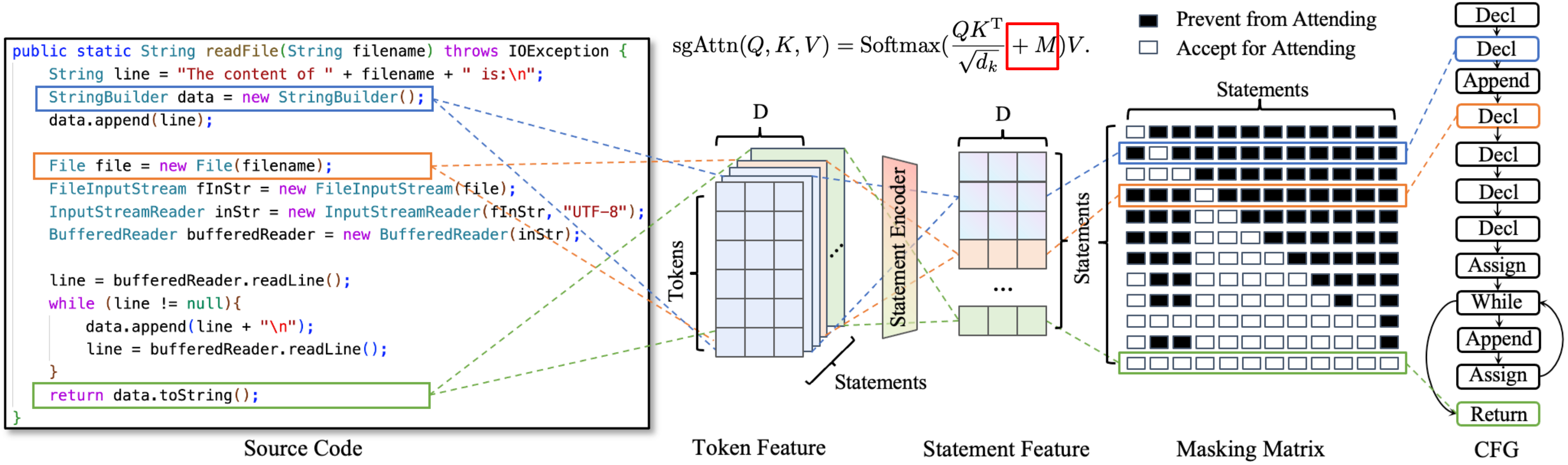


逐步传递机制中易受到无关语句的影响

挑战2：如何避免无关语句的干扰？

主要贡献

- 指出代码中**存在远距离依赖关系(long-distance dependency)**，并且在捕获远距离依赖关系时要**避免引入无关的信息**
- 提出了**受结构指导的自注意力机制(structural-guided attention, *sgAttention*)**

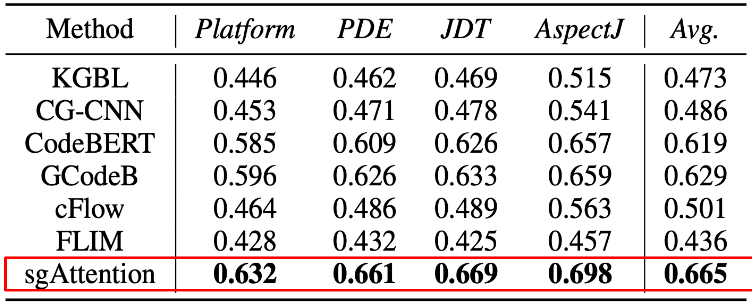


```
public static String readFile(String filename) throws IOException {
    String line = "The content of " + filename + " is:\n";
    StringBuilder data = new StringBuilder();
    data.append(line);

    File file = new File(filename);
    FileInputStream fInStr = new FileInputStream(file);
    InputStreamReader inStr = new InputStreamReader(fInStr, "UTF-8");
    BufferedReader bufferedReader = new BufferedReader(inStr);

    line = bufferedReader.readLine();
    while (line != null){
        data.append(line + "\n");
        line = bufferedReader.readLine();
    }
    return data.toString();
}
```

$$\mathrm{sgAttn}(Q,K,V)=\mathrm{Softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\boxed{+M}\right)V.$$

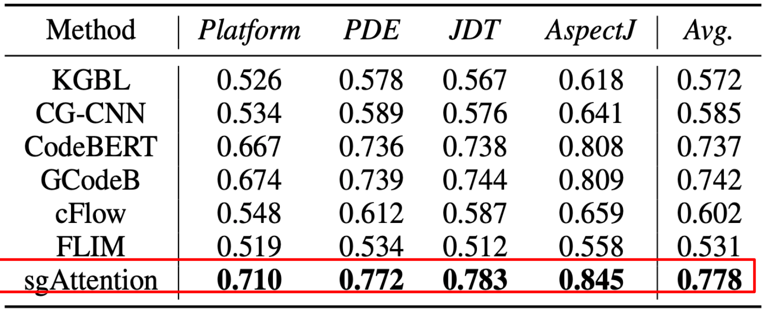
实验效果

- 指出代码中**存在远距离依赖关系(long-distance dependency)**，并且在捕获远距离依赖关系时要**避免引入无关的信息**
- 提出了**受结构指导的自注意力机制(structural-guided attention, *sgAttention*)**
- 在缺陷定位问题中表现超越了state-of-the-art baselines



Method	*Platform*	*PDE*	*JDT*	*AspectJ*	*Avg.*
KGBL	0.446	0.462	0.469	0.515	0.473
CG-CNN	0.453	0.471	0.478	0.541	0.486
CodeBERT	0.585	0.609	0.626	0.657	0.619
GCodeB	0.596	0.626	0.633	0.659	0.629
cFlow	0.464	0.486	0.489	0.563	0.501
FLIM	0.428	0.432	0.425	0.457	0.436
sgAttention	**0.632**	**0.661**	**0.669**	**0.698**	**0.665**

Table 2: The performance evaluation in terms of MAP, and the best performance is boldfaced. KGBL and GCodeB are shorts for KG-BugLocator and GraphCodeBERT, respectively.



Method	*Platform*	*PDE*	*JDT*	*AspectJ*	*Avg.*
KGBL	0.526	0.578	0.567	0.618	0.572
CG-CNN	0.534	0.589	0.576	0.641	0.585
CodeBERT	0.667	0.736	0.738	0.808	0.737
GCodeB	0.674	0.739	0.744	0.809	0.742
cFlow	0.548	0.612	0.587	0.659	0.602
FLIM	0.519	0.534	0.512	0.558	0.531
sgAttention	**0.710**	**0.772**	**0.783**	**0.845**	**0.778**

Table 3: The performance evaluation in terms of MRR, and the best performance is boldfaced. KGBL and GCodeB are shorts for KG-BugLocator and GraphCodeBERT, respectively.

谢谢！