CSBSE 2018

第七届中国基于搜索的软件工程研讨会

地点： 北京化工大学（东校区）会议中心多功能厅

日期： 2018 年 11 月 17 日（星期六）

时间： 08:00 - 17:00

代码自动生成与推荐的几种技术思路

彭鑫

复旦大学

pengxin@fudan.edu.cn

http://www.se.fudan.edu.cn

http://bigcode.fudan.edu.cn

智能化软件开发

人机（AI）结对编程

智能化的编程助手持续观察开发人员的行为和任务上下文，并以推荐、问答、信息可视化等方式在需要的时候提供智能化支持

https://www.codota.com

当前IDE中的代码推荐与补全

*Main.java

```
package pers;

import java.util.HashMap;

import javax.swing.JPanel;

public class Main {

    public void init(){
        JPanel parent = new JPanel();
        |
    }


}
```

*Wizard.java

```
package pers;


public class Wizard {



}
```

*Main.java

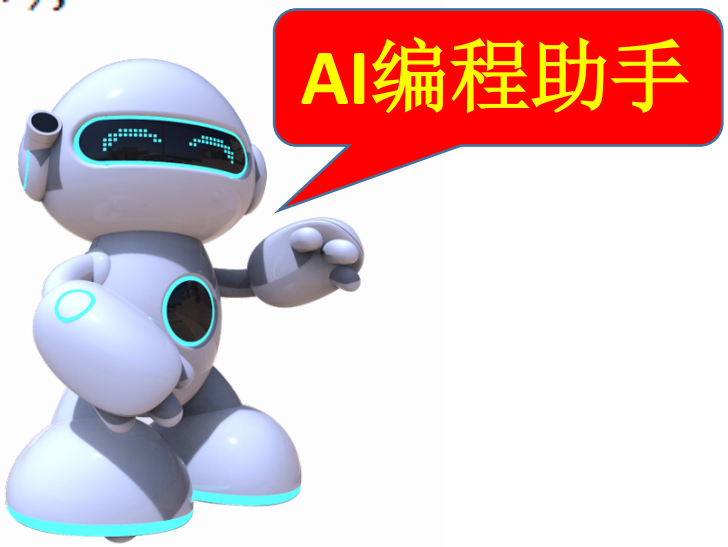
```
package pers;

import java.util.HashMap;

import javax.swing.JPanel;

public class Main {

    public void init(){
        JPanel parent = new JPanel();
        parent.setLayout(null);
    }

    public HashMap list(){
        |
    }

}
```

*Wizard.java

```
package pers;

import java.util.HashMap;

public class Wizard{

    public void init(){

    }


}
```

智能化的代码生成与推荐

```java
import java.io.*;
public class CopyLines {
    public void test()    {
        BufferedReader inputStream=null;
        PrintWriter outputStream=null;
        FileReader reader = new FileReader("../README.md");
        FileWriter writer = new FileWriter("../readme.md");
        inputStream=new BufferedReader(reader);
        outputStream=new PrintWriter(writer);



        if(inputStream!=null) {
            inputStream.close();
        }
        if(outputStream!=null) {
            outputStream.close();
        }
    }
}
```



规模：语句及代码段级的生成与推荐

内容：通用API使用、算法实现、应用逻辑实现

交互：意图理解、问题解释、反馈调整

人工编码过程

需求

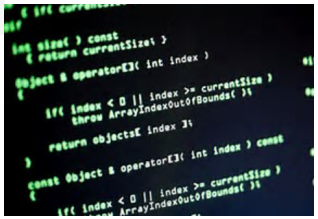

```java
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
public class anoncode{
    //This code is from Head First Java Book page 666 and tests
    //The use of anonymous class definitions
    //which arise in GUI Action Listener situations
    public static void main(String[] arg){
        JFrame frame = new JFrame("This is the Anonymous Class actionListener Code Example");
        final JButton button = new JButton("Click Me");
        //frame.getContentPane().add(button); replaced by next 4 lines
        JPanel pane = new JPanel();
        pane.setLayout(new FlowLayout(FlowLayout.CENTER,20,20));
        pane.setBounds(10, 10, 300, 200); // For some unwarned of reason this sizing  is ignored
        //These lines are at the Heart of Darkness in Java GUI code
        pane.add(button);
        frame.getContentPane().add(pane);
        //Here comes the anonymous code
        //in effect we are declaring a blank action listener class
        //including the methods used to handle our button

        frame.setVisible(true);
    }
}
```

代码上下文

测试用例

实现代码

```java
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
public class anoncode{
    //This code is from Head First Java Book page 666 and tests
    //The use of anonymous class definitions
    //which arise in GUI Action Listener situations
    public static void main(String[] arg){
        JFrame frame = new JFrame("This is the Anonymous Class actionListener Code Example");
        final JButton button = new JButton("Click Me");
        //frame.getContentPane().add(button); replaced by next 4 lines
        JPanel pane = new JPanel();
        pane.setLayout(new FlowLayout(FlowLayout.CENTER,20,20));
        pane.setBounds(10, 10, 300, 200); // For some unwarned of reason this sizing  is ignored
        //These lines are at the Heart of Darkness in Java GUI code
        pane.add(button);
        frame.getContentPane().add(pane);
        //Here comes the anonymous code
        //in effect we are declaring a blank action listener class
        //including the methods used to handle our button
        button.addActionListener( new ActionListener(){
            public void actionPerformed(ActionEvent evt){
                button.setText("Hello Dear");
                button.setBounds(10, 10, 200, 50);
                System.out.println("We can add in our code here");
                System.out.println("Perhaps a close database or Save file ro
            }
        frame.setVisible(true);
    }
}
```



参照代码
（开源、企业代码库）

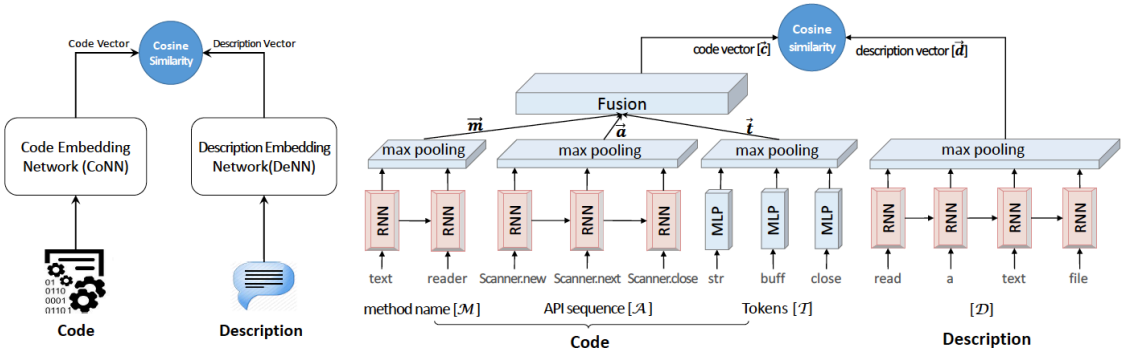
JavaDoc stackoverflow

技术文档及问答知识

几种技术思路

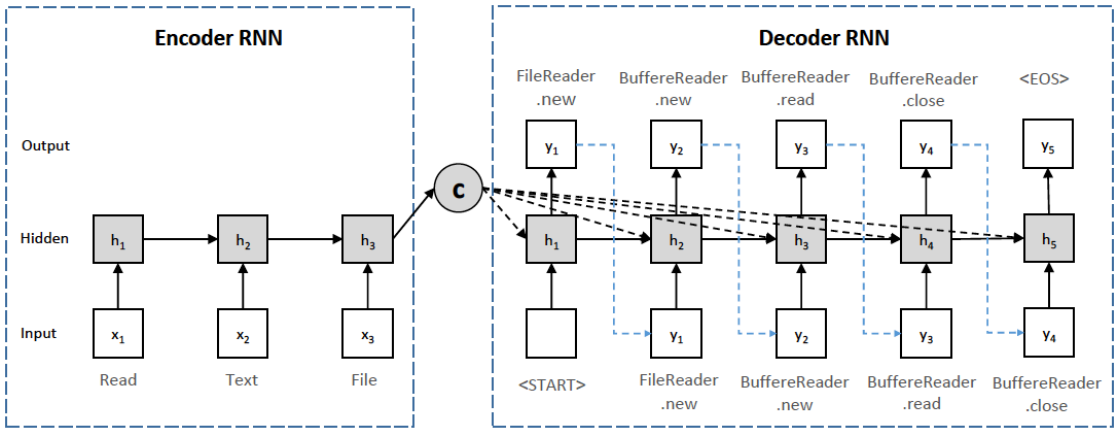
	功能描述	代码上下文	输入/输出样例
信息检索	基于文本的代码检索	基于上下文相似度的代码补全	
模式挖掘		上下文敏感的代码模式匹配	
优化搜索			基于优化搜索的程序合成
统计模型		基于上下文概率统计的代码补全	
深度学习	基于翻译模型的代码推荐	基于上下文特征学习的代码补全	基于深度神经网络的程序合成

基于功能描述的API和代码推荐



通过余弦相似度计算对文本描述和代码特性（方法名、API序列、token）的向量表示的联合训练

Xiaodong Gu, Hongyu Zhang, Sunghun Kim: Deep code search. ICSE 2018: 933-944



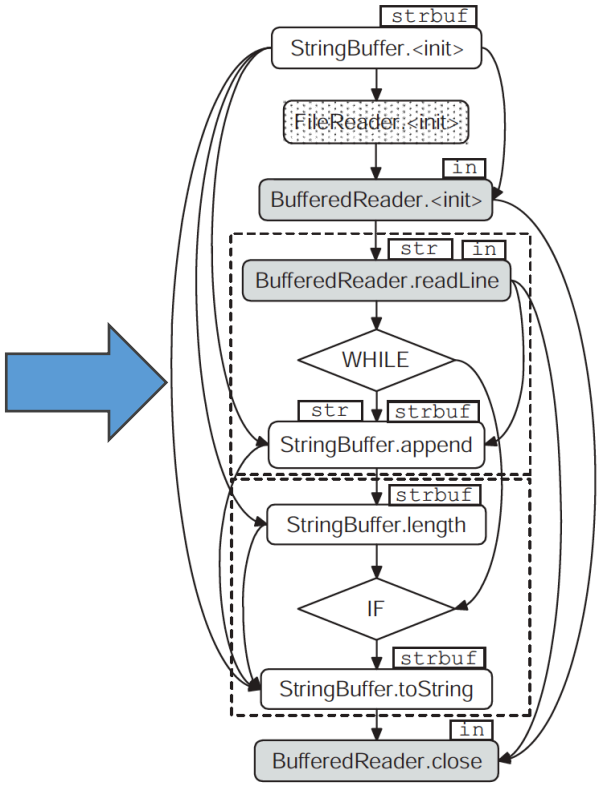
通过训练一个Encoder-Decoder模型实现对于给定功能描述语句的API序列推荐

Xiaodong Gu, Hongyu Zhang, Dongmei Zhang, Sunghun Kim: Deep API learning. SIGSOFT FSE 2016: 631-642

不考虑代码上下文，所推荐的API序列或代码片段未进行实例化及上下文融合

上下文敏感的代码模式匹配

```
StringBuffer strbuf = new StringBuffer();
BufferedReader in = new
          BufferedReader(new FileReader(file));
String str;
...
while ((str = in.readLine()) != null) {
    ...
    strbuf.append(str + "\n");
    ...
}
...
if (strbuf.length() > 0)
   outputMessage(strbuf.toString(),...);
in.close();
```



在基于图挖掘得到的API使用模式基础上，通过抽取代码上下文中的API序列和token等方面的特征匹配候选的API使用模式

Anh Tuan Nguyen, Tung Thanh Nguyen, Hoan Anh Nguyen, Ahmed Tamrawi, Hung Viet Nguyen, Jafar M. Al-Kofahi, Tien N. Nguyen: Graph-based pattern-oriented, context-sensitive source code completion. ICSE 2012: 69-79

仅能推荐API使用模式，且上下文匹配能力较弱

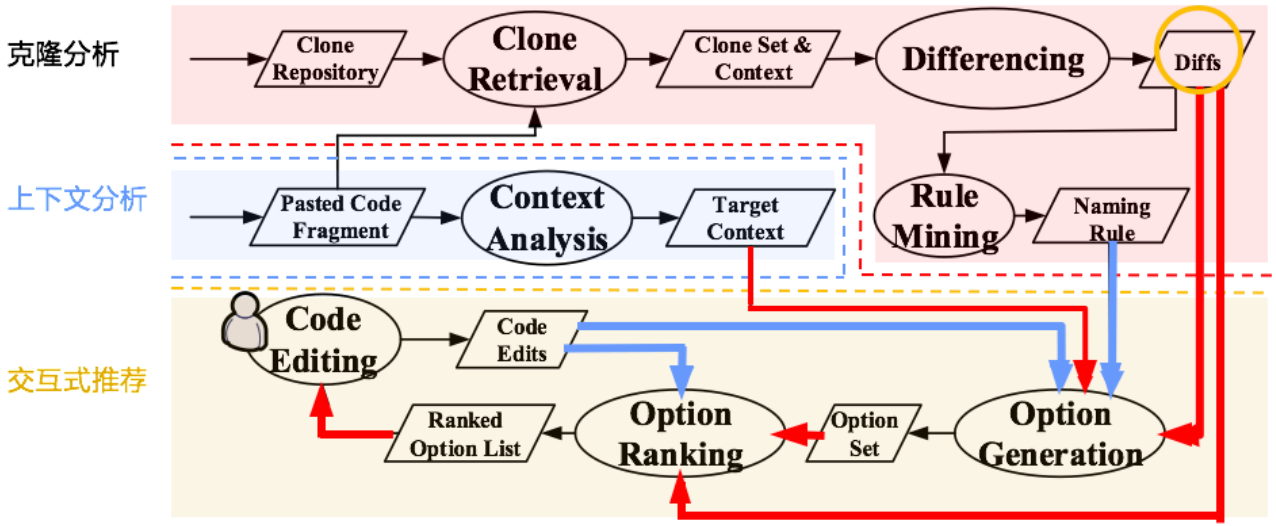
基于克隆分析的代码修改推荐

*CusDOMOutput.java JavaxDOMOutput.java

```
package org.jhotdraw.xml;

import java.io.IOException;

public class CusDOMOutput implements DOMOutput {

    private XMLElement element;
    private String doctype;


    @Override
    public void setDoctype(String doctype) {
        this.doctype = doctype;

    }

    @Override
    public void openElement(String tagName) {

    }

    @Override
    public void closeElement() {
```

在开发人员复制代码时对相关克隆副本进行差异分析，挖掘代码修改模板，在此基础上进行交互式的代码修改推荐

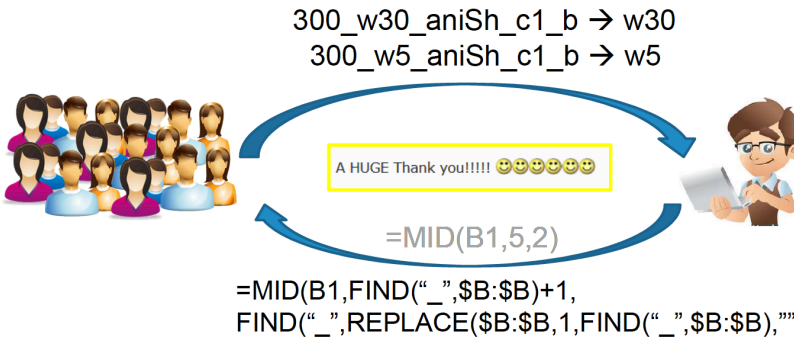
修改点

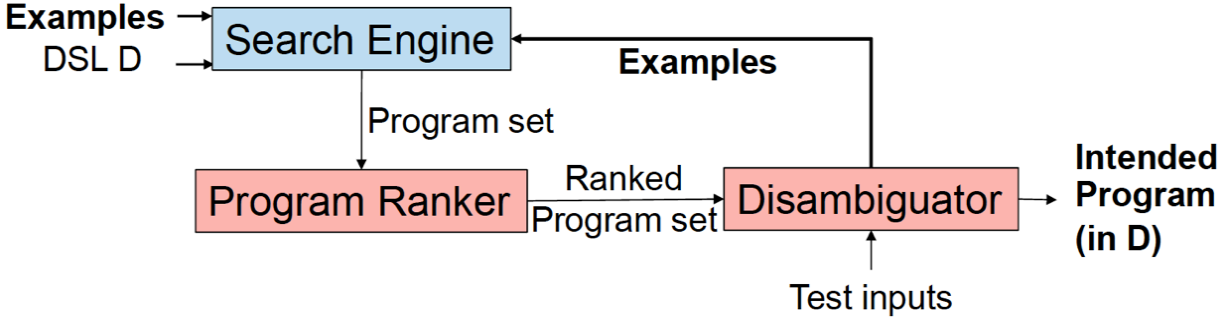


未考虑模板自动匹配，修改建议缺少解释

Yun Lin, Xin Peng, Zhenchang Xing, Diwen Zheng, Wenyun Zhao: Clone-based and interactive recommendation for modifying pasted code. ESEC/SIGSOFT FSE 2015: 520-531

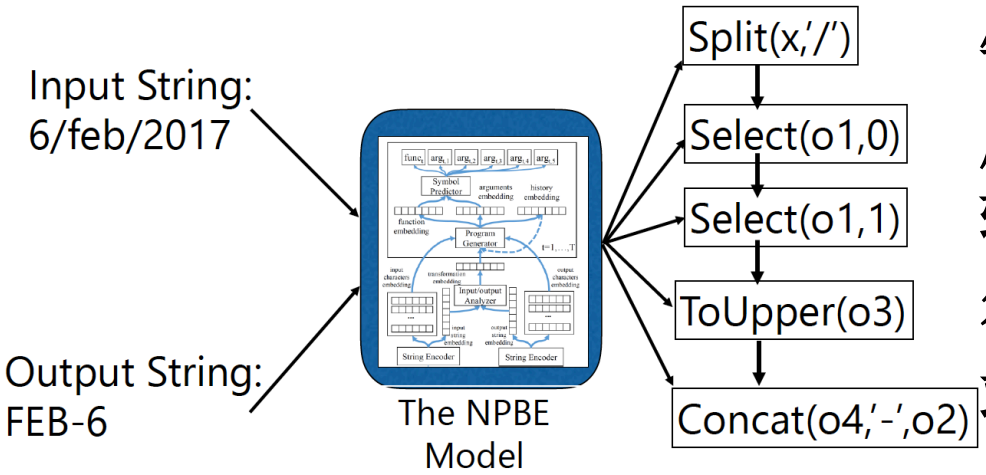
基于输入/输出样例的代码合成





在由**DSL**规范的搜索空间内，利用逻辑推理进行约减，利用机器学习进行排序，通过额外的输入和程序执行信息消除歧义

Sumit Gulwani, Prateek Jain: Programming by Examples: PL Meets ML. APLAS 2017: 3-20



针对字符串操作程序合成的深度神经网络（字符串编码器、输入/输出分析器、程序生成器、符号选择器），可以实现端到端的程序合成

Chengxun Shu, Hongyu Zhang: Neural Programming by Example. AAAI 2017: 1539-1545

局限于面向数据（数值、字符串）处理的小程序

基于上下文概率统计的代码补全 *n*-gram模型

将代码堪称token序列（串），同时假设一段代码中第*n*个token的预测只依赖于前*n*-1个token

$$P(t_0 \dots t_M) = \prod_{m=0}^{M} P(t_m | t_{m-1} \dots t_{m-n+1})$$

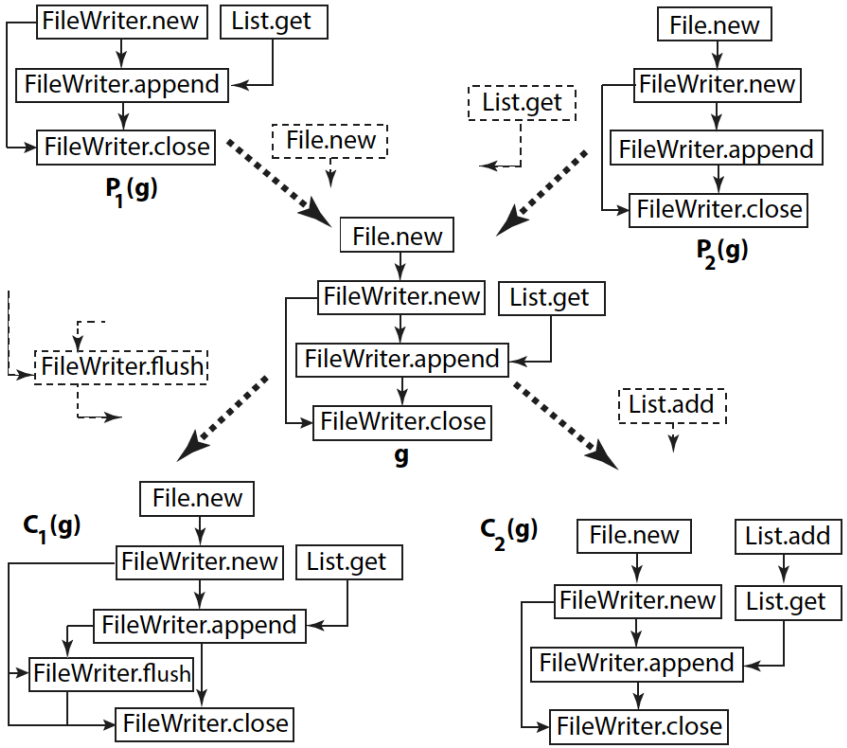
$$P(t_m | t_{m-1} \dots t_{m-n+1}) = \frac{c(t_m \dots t_{m-n+1})}{c(t_{m-1} \dots t_{m-n+1})}$$

Miltiadis Allamanis, Charles A. Sutton: Mining source code repositories at massive scale using language modeling. MSR 2013: 207-216

仅能考虑非常近的代码上下文、考虑代码的文本全序、难以分离混杂的通用代码与特定项目代码

基于上下文概率统计的代码补全

统计图模型



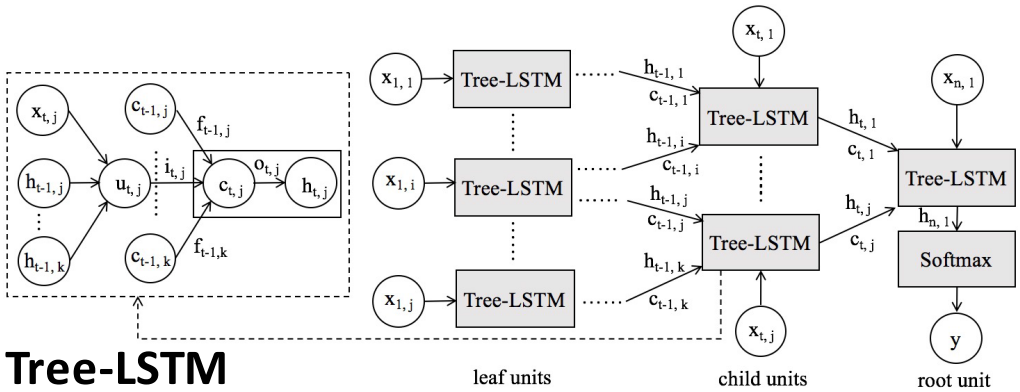
基于API调用、控制单元及其数据流/控制流关系建立图模型

在API上下文图基础上，计算基于给定子图的候选API概率

Anh Tuan Nguyen, Tien N. Nguyen: Graph-Based Statistical Language Model for Code. ICSE 2015: 858-868

仅支持通用API调用及控制单元的预测，合成能力较弱

基于上下文特征学习的代码补全



Tree-LSTM

通过在完整代码中挖除不同长度的代码片段来构造训练样本，定义代码的树状表示并作为输入对深度学习模型进行训练

```
try {
    String host = "jdbc:derby://localhost:1527/Employees";
    String uName = "admin";
    String uPass = "admin";
    Connection con = DriverManager.getConnection(host, uName, uPass);

    Statement stmt = con.createStatement();
    String sql = "SELECT * FROM Workers";
    ResultSet rs = stmt.executeQuery(sql);

    rs.next();
    int id_col = rs.getInt("ID");
    String first_name = rs.getString("First_Name");
    String last_name = rs.getString("Last_Name");
    String job = rs.getString("Job_Title");

    String p = id_col + " " + first_name + " " + last_name + " " + job;
    System.out.println(p);
}
catch ( SQLException err ) {
    System.out.println( err.getMessage( ) );
}
```

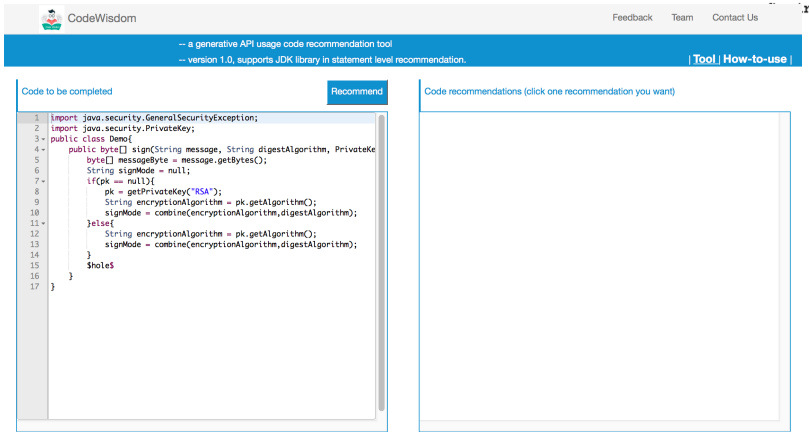
原始代码

训练样本构造

上下文代码

生成代码

```
public byte[] sign(String message, String digestAlgorithm,
   PrivateKey pk) throws GeneralSecurityException {
    byte[] messageByte = message.getBytes();
    String signMode = null;
    if(pk == null){
        pk = getPrivateKey("RSA");
        String encryptionAlgorithm = pk.getAlgorithm();
        signMode = combine(digestAlgorithm, encryptionAlgorithm);
    }else{
        String encryptionAlgorithm = pk.getAlgorithm();
        signMode = combine(digestAlgorithm, encryptionAlgorithm);
    }
}
```



在线演示：

http://bigcode.fudan.edu.cn/CodeRecommendation/index.html

仅支持通用API调用及控制单元的预测，不支持算法代码及特定应用逻辑的补全推荐

深度学习是银弹吗？可能不是！

1）需求的模糊性以及开发者意图的开放性
2）软件项目业务和技术领域的多样性
3）软件代码数据的质量问题

Code Tangling

DB Access　File Access

```
public void insertVocabularyIntoDB(String path) {
   try {
      Class.forName(driver);
      conn = DriverManager.getConnection(url,
           user, password);
      if (!conn.isClosed()) {
         statement = conn.createStatement();
      } else {
         System.out.println("defeated");
      }
      BufferedReader br = new BufferedReader(new FileReader(new
File(path)));
      StringBuffer sb = new StringBuffer();
      String temp = null;
      while ((temp = br.readLine()) != null) {
         String sql = "insert into " + table + " values ('" +
temp.toString() + "')";
         statement.executeUpdate(sql);
      }
      statement.close();
      conn.close();
   } catch (Exception e) {
      e.printStackTrace();
   } catch (Error e) {
      e.printStackTrace();
   }
}
```

Bot and Trivial Messages

modules/apps/foundation/portal/.gitrepo CHANGED

```
@@ -3,7 +3,7 @@
 ;
 [subrepo]
     cmdver = liferay
-    commit = 2f03e545085c159d922fb9eac9b166ee820a94c0
+    commit = c3d68dbcaaa18c18e76bb46697c52e4d8ec6ffa9
     mode = push
-    parent = ab9bdb710f55453499286b0269f60effb1c38e36
+    parent = a1f017cdfb2581a936418d58405863f0262b47c
     remote = git@github.com:liferay/com-liferay-portal.git
```

Bot Message

Reference Message:
ignore Update ' modules / apps / foundation / portal / .

Message Generated by NMT:
Ignore Update ' modules / apps / foundation / portal / .

CHANGELOG.md DELETED

```
@@ -1,7 +0,0 @@
- # Changelog
- ## 0.1 (2014-02-20)
- Initial public release
- *
```

Trivial Message

Reference Message:
update changelog

Message Generated by NMT:
Updated changelog

Zhongxin Liu, Xin Xia, Ahmed E. Hassan, David Lo, Zhenchang Xing, Xinyu Wang: Neural-machine-translation-based commit message generation: how far are we? ASE 2018: 373-384

需要思考...

如何让程序员理解每一个推荐选项并做出选择？如何让他们相信推荐结果？

如果程序员一行都不会写怎么办？

如何了解程序员的意图？

程序员总是顺序性地思考吗？

有经验的程序员是如何写代码的呢？

拿来主义：大粒度的代码复用

AI 我想为一段消息 message 加上签名，
希望得到的输出是签名后的字节数组。

拿来主义：推荐相似的代码片段甚至模块，指导开发人员进行修改

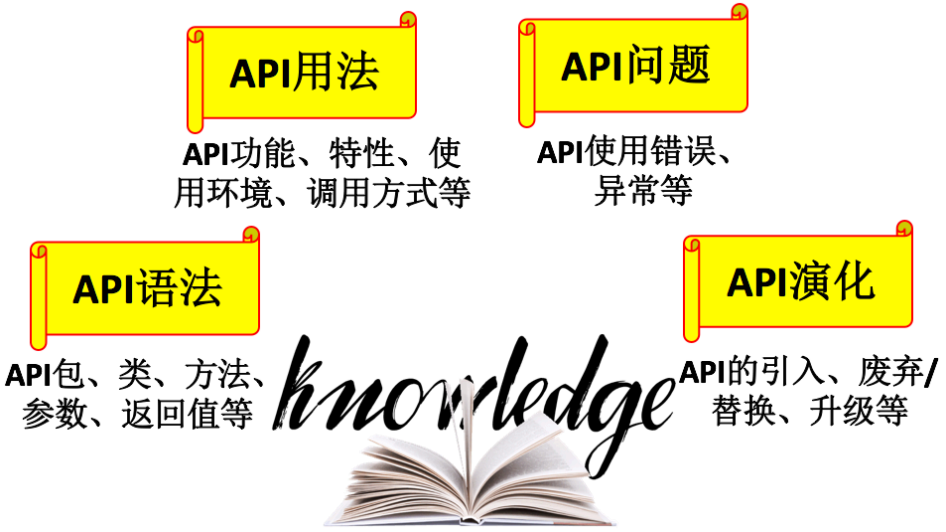
深度学习：在细粒度的代码单元基础上进行训练，然后再逐步合成

理性主义：基于知识的推理

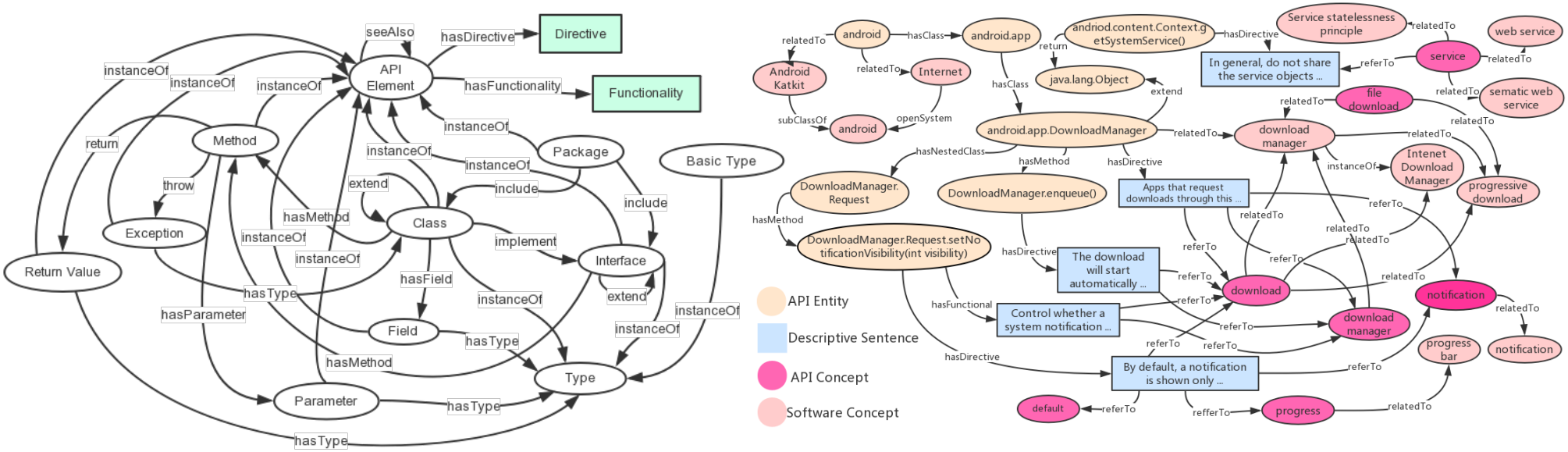
```
public static byte[] sign(String message, String hashAlgorithm, PrivateKey pk)
        throws GeneralSecurityException {
    message = message.toUpperCase();
    byte[] messageByte = message.getBytes();
    String signMode = null;
    signMode = hashAlgorithm + "with";
    if(pk == null){
        pk = getDefaultPK();
        signMode += pk.getAlgorithm();
    }else{
        signMode += pk.getAlgorithm();
    }
}
```

AI 我想要完善这段签名加密的代码

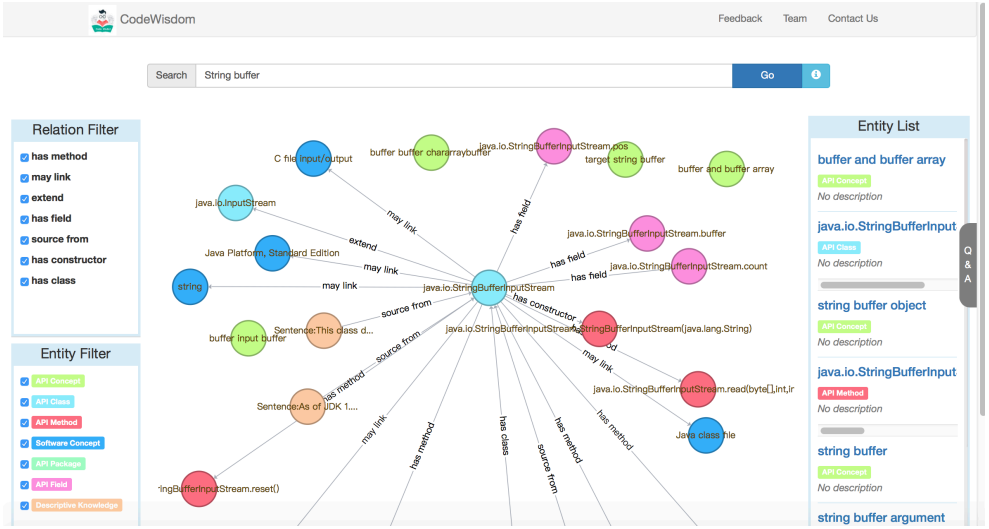
基于意图进行功能分解，确定实现意图的核心API，围绕核心API补充所需代码



API知识图谱



在API元素基本定义结构基础上，通过知识融合实现API描述文本中共性概念的提炼以及与通用知识图谱的链接

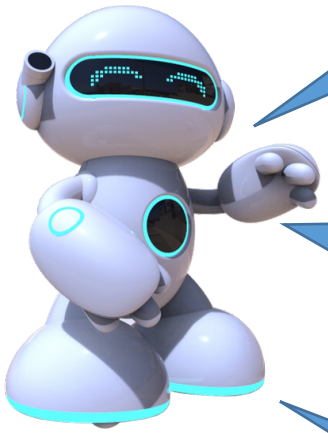


基于知识图谱的关系拓扑实现了初步的精准语义搜索等应用，需进一步完善图谱的形式化程度以提供推理能力

在线演示：http://bigcode.fudan.edu.cn/KnowledgeGraphAPI/index.html

向结对程序员学习

基于知识的交互式澄清与解释

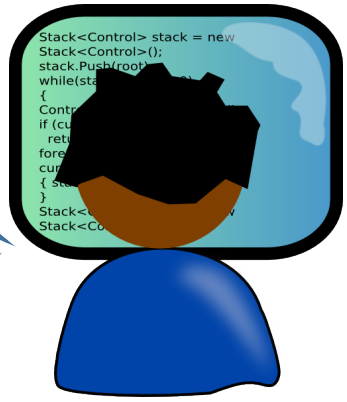


You can use StringBuffer here, as it is thread safe.

Do you want to write the content to a file or print it to the console?

OK. The code is ready. You can change the string variable to specify the file path.

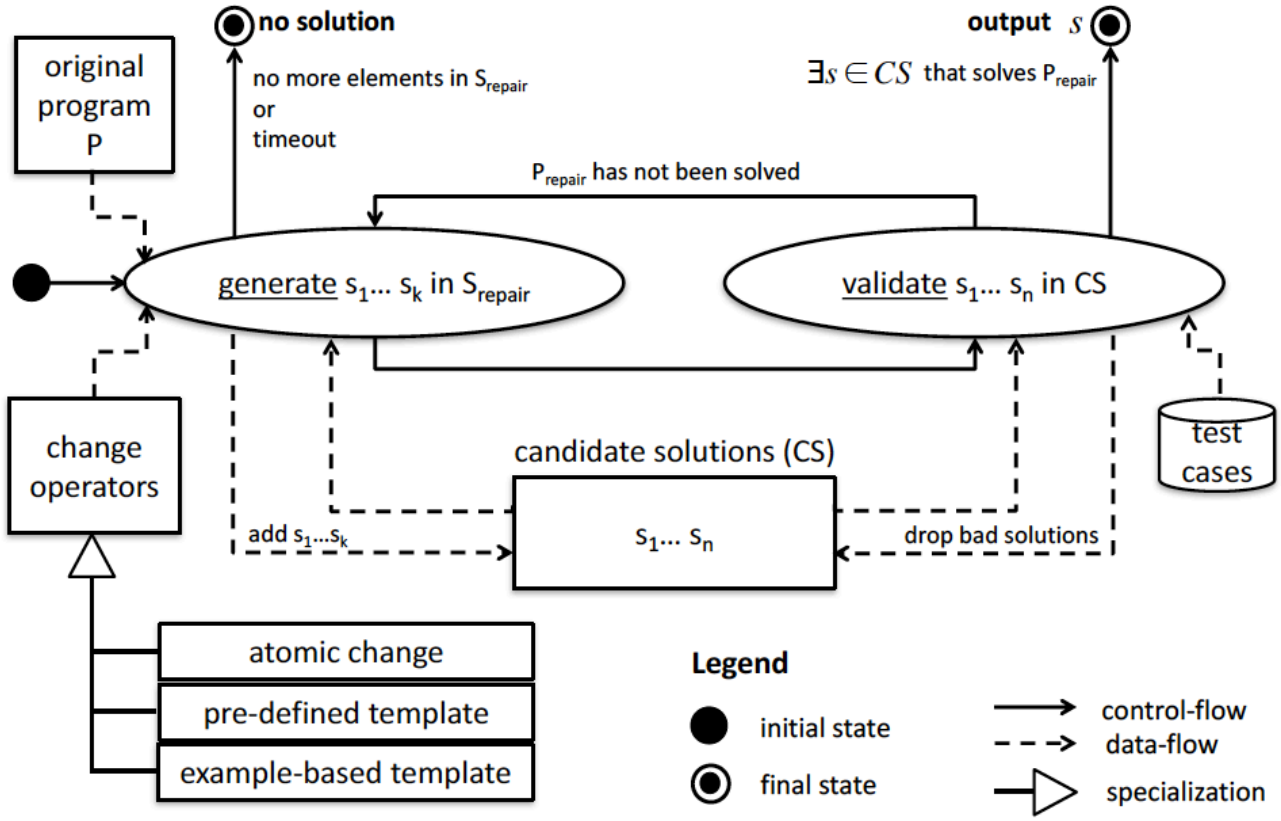
Yes, that is what I want.



Write to a file.

...

基于搜索的程序自动修复



在测试用例基础上，利用基于搜索的程序自动修复实现自动生成代码的完善和校准

Luca Gazzola, Daniela Micucci, Leonardo Mariani: Automatic Software Repair: A Survey. IEEE Transactions on Software Engineering (Early Access).

总结

现状

- 现有方法仅在单一类型的推荐上较为有效，难以兼顾API使用/算法/业务逻辑、通用/特定应用逻辑等不同方面
- 存在意图不明确、领域多样性、数据质量差等根本性困难

展望

- 人机协同的交互式智能化推荐：澄清、解释、试探、反馈
- 知识基础：领域业务知识、通用API知识、算法知识等
- 多形态、多层次的智能化推荐
  - ✓ 重新思考代码大数据背景下的构件组装与定制化开发
  - ✓ 多形态推荐：检索匹配+辅助修改、自动生成
  - ✓ 多层次智能：概率模型探索、知识推理精化、交互澄清意图、搜索完善校准

CSBSE 2018

第七届中国基于搜索的软件工程研讨会

地点： 北京化工大学（东校区）会议中心多功能厅

日期： 2018 年 11 月 17 日（星期六）

时间： 08:00 - 17:00

谢 谢！

彭鑫

复旦大学

pengxin@fudan.edu.cn

http://www.se.fudan.edu.cn

http://bigcode.fudan.edu.cn