阶段二工作安排

在一阶段的基础上，进行一定的预处理优化。涉及范围有：使用rVSM模型改进基础VSM模型，引入堆栈信息，用AST解析代码，查询优化，历史查询

我们需要明确的是这个项目的工作流程，首先对代码进行预处理，然后接下来使用到的一切的一切，都和sc无关了，使用的都是我们预处理好了的文件。然后查询也不是所有一起扔进来，按照时间一个个扔进来

rVSM

用将VSM模型替换为rVSM模型，避免模型对长文件的偏好

结构化信息

对bugReport中切分出SC，NL，ST这三类语素。

NL：自然语言

SC：代码

ST：堆栈

其中NL部分交给原先的查询部分来解决，SC暂定和NL一起处理，ST因为包含大量无关信息，所以直接切除

AST树

使用JavaParser解析SC，切分出其中所有的类名、函数名、变量名、注释，转化为NL进行预处理，把这些与处理完的东西按照文件放在一个txt里。将这些切分出的内容交给词处理方法进行预处理。然后相关的import要有一个单独的记录，给ST匹配使用。

查询优化

增加NL部分停用词的词类，把废物NLTK-stopwords鲨掉，使用效果更好的停用词和领域词，因为目前还不可以进行查询重构，就先这样

历史查询

对于bugReport的匹配工作需要单独抽离开，系统要有一个记录历史查询的功能接口，用来记录bugReport以及他是否已被解决。

对bugReport之间的相似度进行**传统VSM匹配**（因为不是B话越多就越有普适性的），匹配早于这个Bug的所有相似bugReport，然后对相似度大于一个阈值（需要尝试）的**已解决的bug**所对应的代码文件，在匹配时提高权重