

**哈尔滨工业大学**

**DevRec进度报告**

**王世川**

**597944330@qq.com**

17766520412

**日期：2018.04.08**

目录

[DevRec框架： 3](#_Toc510968163)

[总结： 3](#_Toc510968164)

DevRec框架：

基于bug report的分析：

用LDA模型抽取summary和description生成主题，对每份缺陷报告生成：

术语频率统计向量：

（术语1{次数}，术语2{次数}，……，术语n{次数}）

主题分布概率向量

（主题1{概率}，主题2{概率}，……，主题n{概率}）

提取所有的缺陷报告的产品、模块信息，并转化成01表示的向量：

（产品1{0，1}，产品2{0，1}，……，产品n{0，1}）

（模块1{0，1}，模块2{0，1}，……，模块n{0，1}）

以上四组向量合并成一个向量，作为feather，用于MLKNN评估缺陷报告之间的距离

提取修复者列表转化成01表示的的向量：

（开发者1{0，1}，开发者2{0，1}，……，开发者n{0，1}）

该向量为标签。

对一份新的缺陷报告，提取以上提到的特征向量，用MLKNN算出br-score，推荐出一组开发者，每个人有一个br-score分数。

基于developer的分析：

利用上面已有的特征矩阵，统计每个开发者的主题亲和分数，按论文说的公式，计算出的是该开发者对每个话题的参与度的乘积，反映的是活跃度：

此处有个疑惑，当数据集较小的时候，每个文本的话题预测总能覆盖所有的话题，则此时该分数按照公式与新的缺陷报告并无关系，我认为将新的缺陷报告的主题概率分布代入公式能增加准确性。

利用新缺陷报告的产品、模块、计算每个开发者在这些产品、模块中的参与程度，得到产品亲和分数（参与涉及此产品的所有缺陷报告中的多少）、模块亲和分数（参与涉及此模块的所有缺陷报告中的多少）。

提取每个开发者发表过的缺陷报告的所有术语中的高频前十个词语，提取新报告的术语频率统计，两者计算可以得出术语亲和分数

最终通过以上五个分数，建立DevRec框架，用百分之十的训练实例，代入估算出五个分数的权重比例，估算使用随机贪婪方法，不一定是最优权重。

总结：

清明节假期利用空闲时间实现了标绿色星号部分的工作，其他工作借鉴之前的基础无需太多时间。

通过之前的工作和反复阅读该论文，已经完全了解DevRec方法的步骤。

前两个礼拜参加了四个比赛以及暑期实习工作的笔试，时间上多有耽搁请见谅。