

**哈尔滨工业大学**

**DevRec进度报告**

**王世川**

**597944330@qq.com**

17766520412

**日期：2018.04.15**

目录

[DevRec缺陷报告分派方法： 3](#_Toc511668382)

[I. 需求： 3](#_Toc511668383)

[II. 进度： 3](#_Toc511668384)

[总结： 5](#_Toc511668385)

DevRec缺陷报告分派方法：

1. 需求：

围绕缺陷报告修复者推荐 讨论目前完成的工作，遇到的问题，解决的方法，我们要做的论文的思路，创新点等，请各位同学提前简单整理一下现在研究的内容，以及目前遇到的问题，方便明天讨论

1. 进度：
   1. 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（去掉出现次数较少的术语）：

（术语1{次数}，术语2{次数}，……，术语n{次数}）

主题分布概率向量（当话题数较多的时候，n<topic\_number）:

（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矩阵**X**，作为feature，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：（去掉出现次数较少的开发者）

（开发者1{0，1}，开发者2{0，1}，……，开发者n{0，1}）

该向量为标签**y**。

* 对一份新的缺陷报告new-br，提取以上提到的特征向量，这些特征向量可以用词袋和LDA模型自动生成，用MLKNN算出br-score，推荐出一组开发者，每个人有一个br-score分数。
  1. 基于developer的分析：
* 术语亲和分数：

对每个开发者统计其参与过的br的术语总量，这一过程既（**y矩阵的转置乘以术语向量矩阵**）。对结果只保Top-10，既将其他非topic10词语频率置0（**x[x<top10]=0**），该矩阵记为dl-term，每个开发者用词总量为Nd，相当于（**dl-term矩阵乘以n行一列的全一矩阵**）。得到Nd后对dl-term大于0的频率保留1做记号（**x[x>0]=1**）

对br-new的术语统计总的术语次数，为Nb，计算方法同上。Br-new中开发者Top-10列表中的术语出现的次数为**Nbd(br-term乘以（dl-term的转置）)，**则开发者对这份报告的术语亲和分数为：

**Term-score = Nbd/（-Nbd +Nd.T+Nb）**

* 主题亲和分数：

对每个开发者统计其参与过的br的主题将这些概率相加，除以所有报告这一概率的和，这一过程既（（**y矩阵的转置乘以主题频率向量矩阵），除以（全一矩阵乘以主题频率向量矩阵**）），记为dl-topic矩阵（行为开发者）。

对br-new所有主题频率不为0的主题置1，则开发者对这份报告的术语亲和分数为：（此处不为零置1可能没有很好的体现这个报告的主题倾向）

**Topic-score = 1 - （br-topic 乘以（1- dl-topic.T））**

* 产品亲和分数：

对每个开发者统计其参与过的产品的次数，除以这些产品出现的总次数，既：（y的转置乘以product矩阵），除以（全一矩阵乘以product矩阵）。记为dl-product矩阵

对new-br的产品矩阵br-product有：

Product-score = br- product乘以 dl-product.T

* 模块亲和分数：

对每个开发者统计其参与过的模块的次数，除以这些模块出现的总次数，既：y的转置乘以component矩阵，除以（全一矩阵乘以component矩阵）。记为dl-cp矩阵

对new-br的产品矩阵br-cp有：

Product-score = br-cp 乘以 dl-ccp.T

* 1. DevRec：

对以上五个分数，以一组范围在[0-1]的参数代入计算最终的推荐分数：

Score = gama1\*br\_score + gama2\*term\_score + gama3\*topic\_score + gama4\*product\_core + gama5\*component\_score

，

**评估标准**采用了recall@5和recall@10，既推荐五人中真实开发者比例，和推荐十人中真实开发者的比例。

**评估方法**如下：

先将数据集11等分，第一部分训练、第二部分测试，第一二部分训练、第三份测试，第一二三部分训练、第四份测试……

得到的十次结果取平均值。

参数gama采用随机贪婪算法用十分之一的训练材料估算。

**在最后对实现性能的检验过程中，我所做的程序得到的recall@5和recall@10比论文所做的实验的效果差，具体原因正在排查。具体实验数据将在答辩时给出。**

* 1. 超参数：

本次论文中用到了大量超参数，这些数字很多没有经过验证、调参，这些参数将是下一步工作的重点。

词频下限：10 or 20

开发者出现次数下限：10

LDA：alpha、eta、迭代次数：default to num\_topic/50，0.01，500

K近邻：default to 15

开发者亲近术语top-k：default to 10

一组比例参数gama

**已经做的成果其中值得关注的是，对应术语亲和分数的比例的gama值，在多次验证中都几乎为0，说明此数据的关联性比较差。**

**具体实验数据将在答辩时给出。**

总结：

本阶段已经完成了DevRec模型，关于结果性能不佳或可能存在的代码错误需要进一步排查。另外对于论文说给出的许多未经验证的超参数值得进一步考量。