软件需求工程LAB1

开源社区SmartIDE的需求排序

学校：南京大学

院系：计算机科学与技术系

姓名：许家铭、黄逸维、陈永健

学号：171860581、171860542、171860568

邮箱：[381026323@qq.com](mailto:381026323@qq.com)

1. 概述

本次实验是针对SmartIDE的需求进行重要度排序工作。通过对开源社区的Smart IDE抓取他们开发和维护过程中的bug report，从中分析获取他们所对应的需求，并进行紧急程度的分类排序。本次实验，我们选择了使用LUIS和Word2Vec两种方法来进行自然语言处理和分类，尽我们所能的完成了本次实验。

1. 实验背景

在日常开发的过程中，每天都会有新的bug被发现，报告出来，但并不是每个问题都同样重要，都需要立即处理。往往只有那些影响大，受众多，影响运维的bug才会被着重安排修复。因此我们就需要对bug进行分级，根据其紧急程度进行排序，从而进行bug修复。

实验任务：

1. 数据获取

从Bugzilla的开源代码中，我们爬取了其中的bug report的数据，根据提交的描述作为我们分类所使用的原始数据。

1. 需求排序

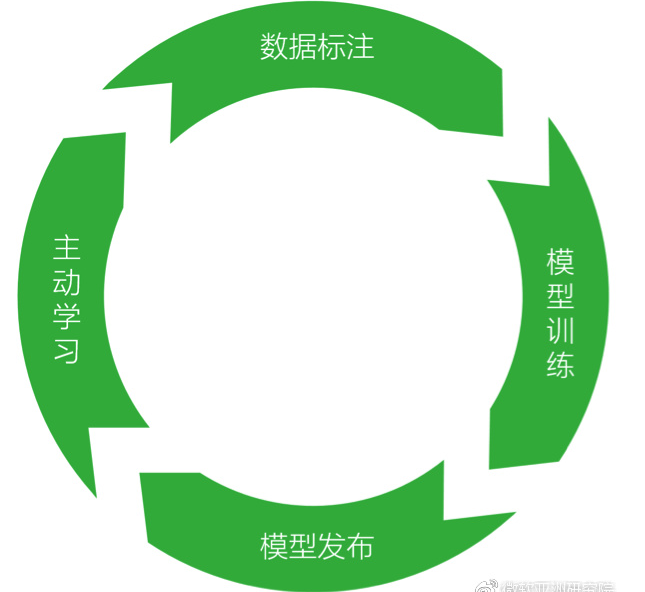
对抓取的数据进行分析，从重要和紧急两个方面对于其进行判别，判断其属于五级的bug中的哪一类bug。

1. 方法验证

根据所使用的需求排序的方法，对所获取的数据进行需求排序，并且与bugzilla上提出的标准的缺陷等级对比，观察准确度。

1. 名词解释
2. LUIS

LUIS (Language Understanding Intelligent Service，https://www.luis.ai) 是微软发布的面向开发者的自然语言语义理解模块开发服务。



LUIS开发流程首先是标注一定的起始数据，后训练得到语义理解模型，再对模型进行必要的测试，发布模型并应用到真实用户场景。根据应用日志继续标注更新模型，使其更加精确。

1. 分类标准

由于我们赚取的数据中，存在大量的P3，其他几种紧急度基本没有，所以我们本次实验着重使用了其五中缺陷描述（blocker、normal、major、minor、enhancement）来辅助我们建立语料库，判别其需求优先度。

语料库的建立：

本次实验中，为了使用NLP对数据进行处理，我们使用了自主建立的语料库。建立的方式主要是通过根据缺陷描述类型和抓取的紧急程度等级，进行人工的贴标签，我们每一种优先级选取一部分例句，放在了语料库中，作为我们分类标准。

3.Word2Vec

Python的一种词转词向量的第三方库，用于将进行过分词处理的词转化为计算机可以使用的向量的形式。接着通过训练进行模型建立。

1. 实验方法

首先，我们使用Python编写了爬虫，爬取了eclipse一周以来的bug report。

其次，我们进行分工贴标签，并进行汇总，得到我们的语料库。

接着我们分为了两组。

一组使用LUIS进行实验，将语料库输入LUIS，建立起NLP模型，并且训练，测试后发布，形成API供我们进行调用。将调用结果返回数值最高的作为我们的处理结果。

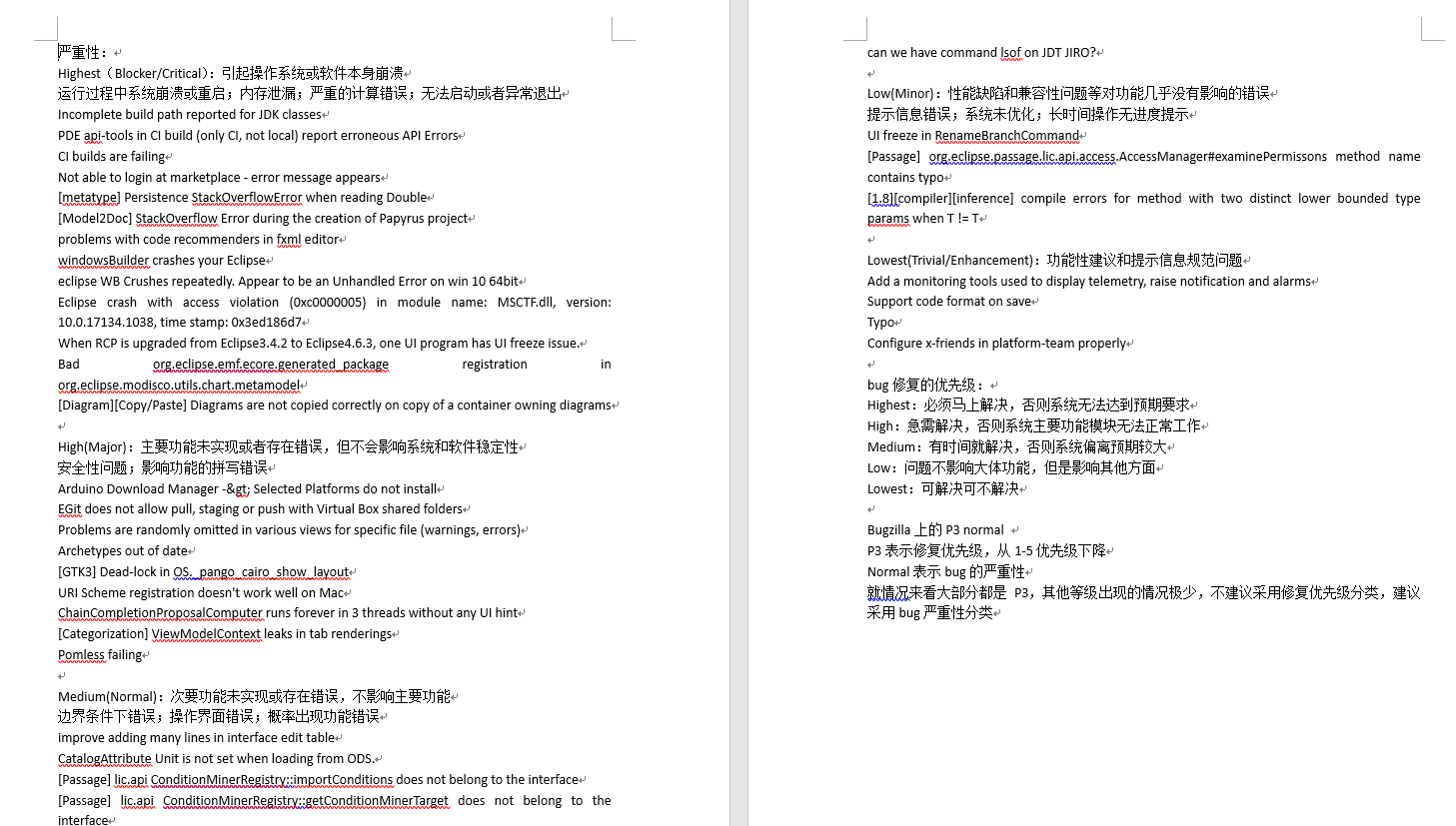
另一组完成了词转词向量，并且用gensim并且进行模型训练，通过单词之间向量距离作为关联程度，对于每个句子，我们将分词后的原始实验数据，每句话与五个等级之间的关联性进行了相加，取出关联性最高的那一个，作为我们的处理结果。

1. 实验过程

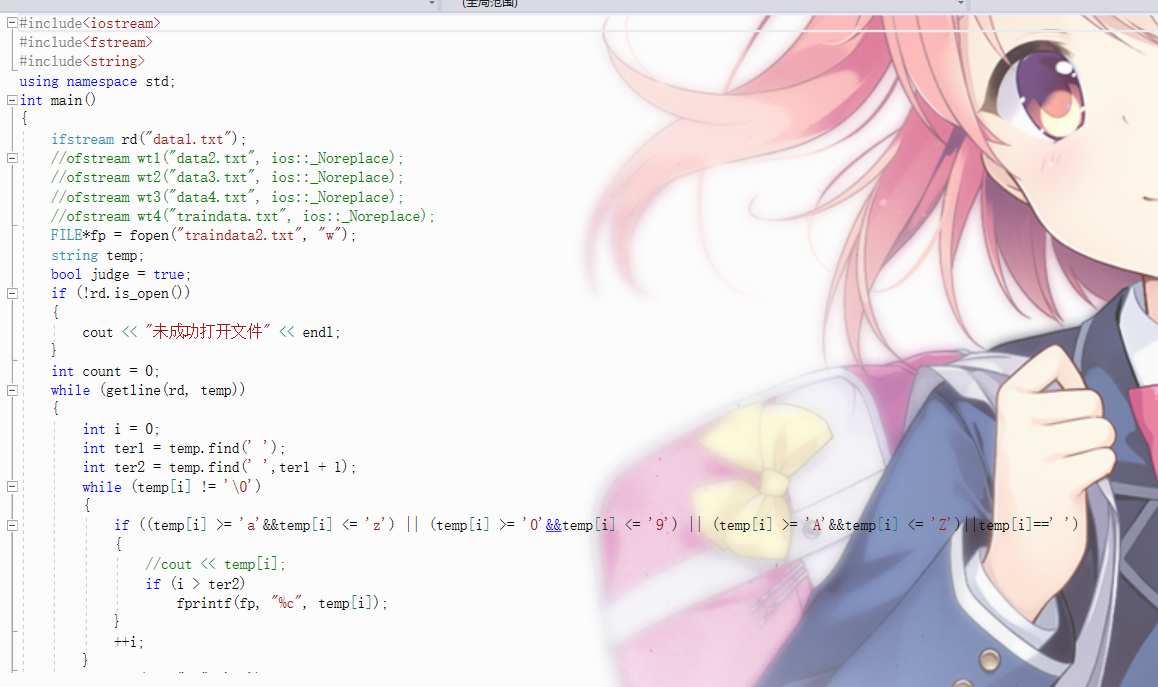
爬虫程序：



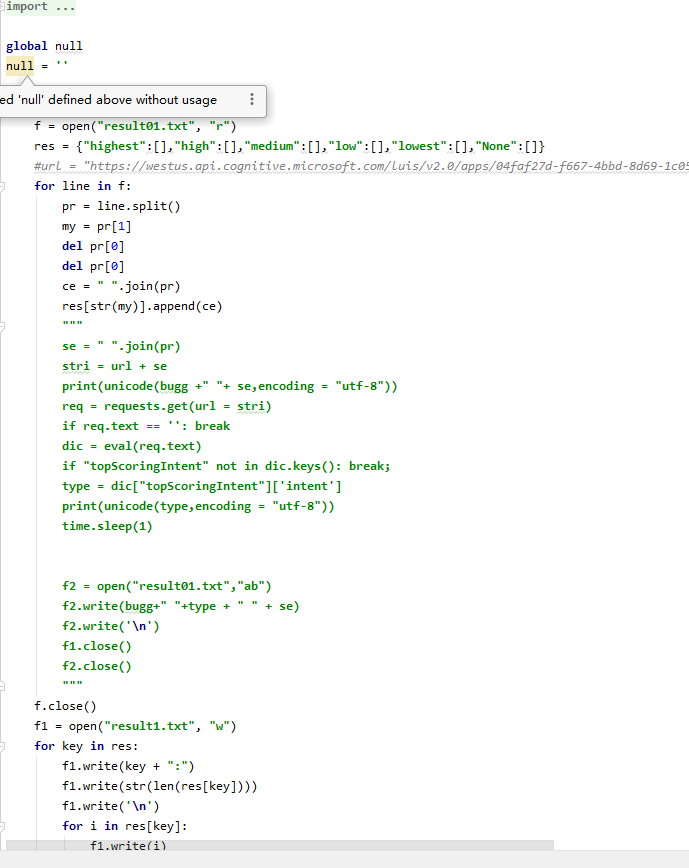
语料库建立：



分词：



调用API的程序：



词转词向量、模型的建立、结果测试运行：



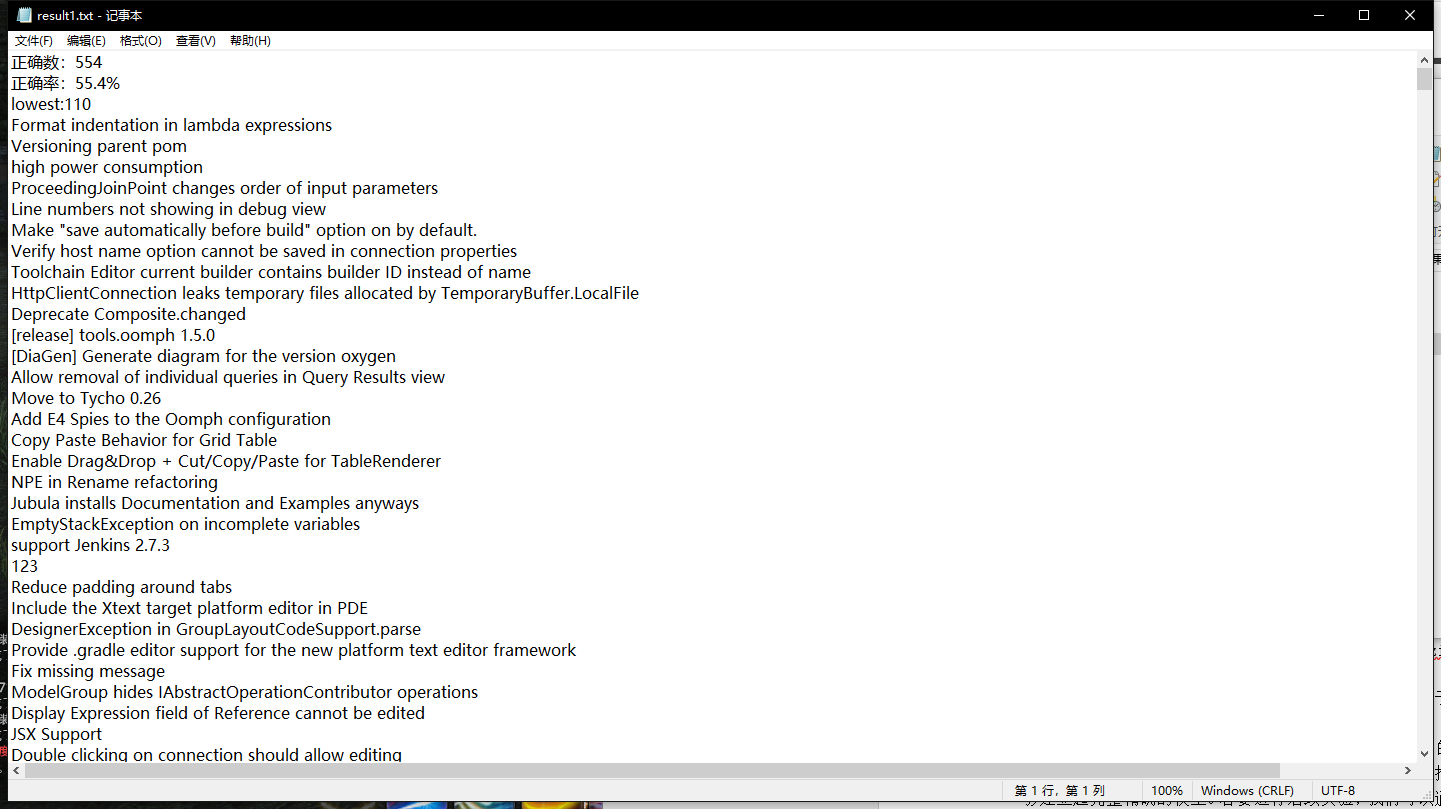
以上使用的所有程序均附在GitHub中

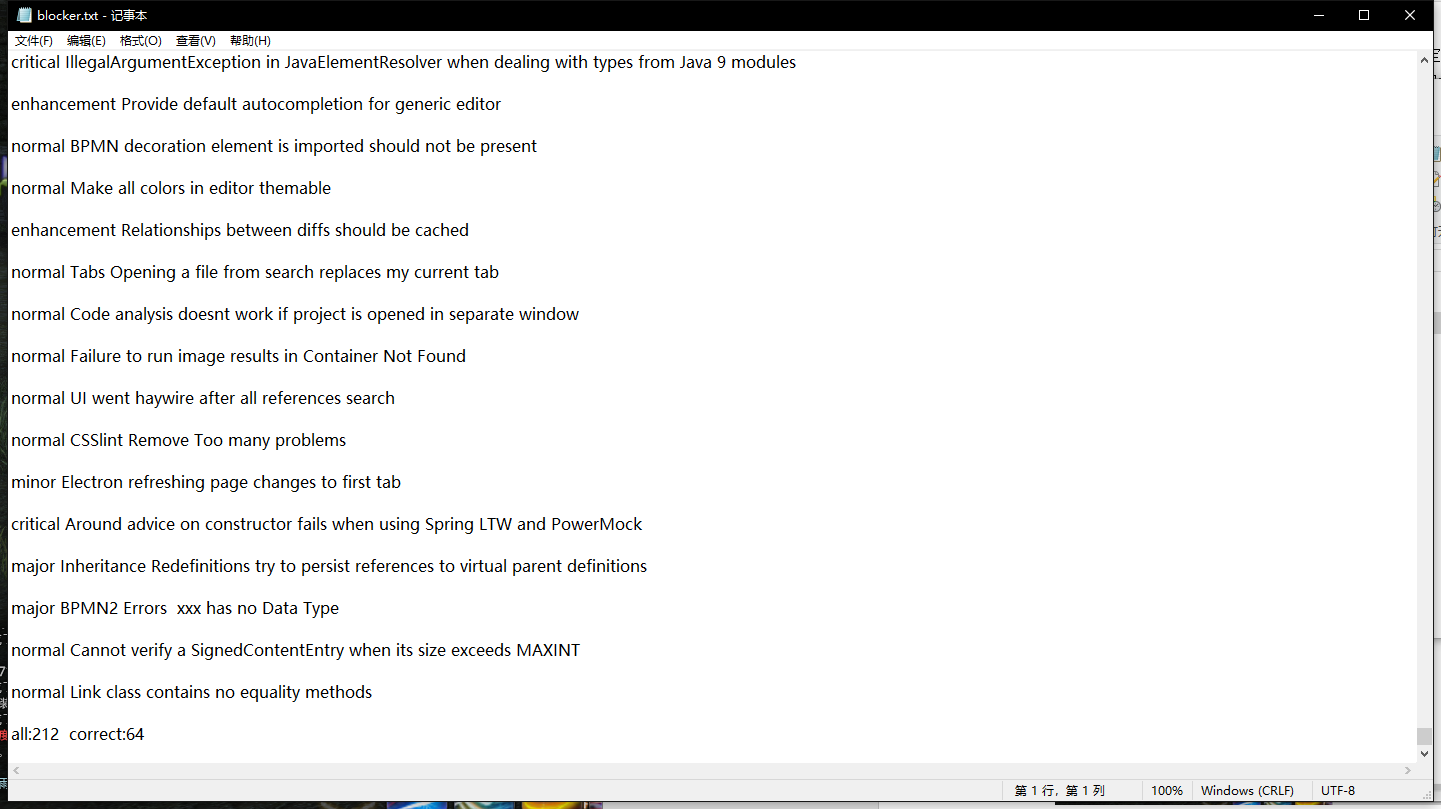
1. 实验结果

根据初步测试的结果来看，分类的精确度还是有一定的。

经过初步检测，LUIS的准确度在50%左右，而我做的词转词向量模型精确度只有25%左右。

结果示意图如下：





具体的结果我已经放入了~/ lab2\_需求优先级/实验结果 中。

1. 结果分析

从我们测试所得的数据来看，我们分类的数据虽然有一定的精确度，但还是和bugzilla上的bug report分级有着一定的差距，不仅仅因为bugzilla使用的是人工投票分级。

这些误差主要来源分别如下：

LUIS:

首先最主要的是，我们的语料库不够完整和全面。很显然，我们自主建立语料库的这种方法是不够的完整而精确的。我们的模型例句不够充足，若要进行后续实验，我们可以通过增加例句的形式来使我们的模型更加精确。

并且我们使用的是每个人各自去找例句分类的方法，相当于只有一个人在进行label的工作，这显然会导致我们贴上的标签不够客观，会有很强的个人理解在其中。如果有更多时间，我们可以通过从中抽取一份样本，每个人给样本进行分类，然后对我们的分类结果进行汇总，如果出现分类结果不同的情况，我们可以通过讨论或者进行投票，请更多人进行评判，从而获得一份更加符合大众价值观的分类标准和语料库。

接着是我们的分类标准不够清晰，由于没有统一的规范，我们查询eclipse的文档的时候，也没有找到官方使用的分类标准，因此我们只能通过对于我们抽选的样本进行了分类，并且我们的分类之间的界定也没能做到很清晰，这一点我们可以通过对于分类进行精确定义来实现。后续也有待改进。

Word2Vec:

这个方法的最大的问题就在于，我没有对分词完的数据进行过滤。由于是期中考试周，我没有来得及在做这个算法的时候，整理出一份过滤的名单，因此有很多无关紧要的词，比如be动词，介词等没有实际意义的词语也被我纳入了统计范围。因此这个结果才会和LUIS有这么大的差距。如果需要改进，我可以通过过滤无实际意义的词语等方式，提高结果准确度。

1. 结论

本次实验，我们使用了微软的NLP模块LUIS和Python的gensim库进行了自然语言处理与需求优先级排序，取得的结果还是比较精确的，但是还是存在一定误差。在后续实验中，我们可以通过增加例句，界定分类标准，多人贴标签汇总等方式进行实验的细化。

在本次实验中，我们组的分数分配为每个人得分1/3。

1. 参考文献

<https://docs.microsoft.com/zh-cn/azure/cognitive-services/luis/what-is-luis>

<https://blog.csdn.net/TYOUKAI_/article/details/77917974>

<https://blog.csdn.net/x_iesheng/article/details/85035149>

<https://zhuanlan.zhihu.com/p/66904318>

<https://bainingchao.github.io/2019/02/13/%E8%87%AA%E7%84%B6%E8%AF%AD%E8%A8%80%E5%A4%84%E7%90%86%E4%B9%8B%E4%B8%AD%E6%96%87%E8%87%AA%E5%8A%A8%E5%88%86%E8%AF%8D/>

<https://stackoverflow.com/questions/4867197/failed-loading-english-pickle-with-nltk-data-load>

<https://blog.csdn.net/HuangZhang_123/article/details/80277793>

<https://blog.csdn.net/churximi/article/details/51472300>

<https://zhuanlan.zhihu.com/p/45207079>

<https://www.cnblogs.com/n2meetu/p/8124267.html>

<https://blog.csdn.net/liugallup/article/details/51164962>

<http://www.ijiandao.com/2b/baijia/243528.html>

<https://cloud.tencent.com/developer/article/1335261>

<https://blog.csdn.net/weixin_44081621/article/details/88781156>

<https://blog.csdn.net/sparkexpert/article/details/80076796>

<https://36kr.com/p/5140965>