开源软件过程模型与改进分析——结巴分词

MipaProject-

1. 软件介绍

我们选择的软件是“结巴分词”，一项开源的 Python 中文分词组件。

GitHub: https://github.com/fxsjy/jieba

中文分词(Chinese Word Segmentation)指的是将一个汉字序列，切分成一个一个单独的词。分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。我们知道，在英文的行文中，单词之间是以空格作为自然分界符的，而中文只是字、句和段能通过明显的分界符来简单划界，唯独词没有一个形式上的分界符，虽然英文也同样存在短语的划分问题，不过在词这一层上，中文比之英文要复杂的多、困难的多。（百度百科-中文分词）

中文分词是文本挖掘的基础，对于输入的一段中文，成功的进行中文分词，可以达到电脑自动识别语句含义的效果。中文分词技术属于自然语言处理技术范畴，对于一句话，人可以通过自己的知识来明白哪些是词，哪些不是词。但如果需要让计算机也能理解，其必需的处理过程就是分词算法。

“结巴分词”就是一项中文分词组件。其 Python 语言实现版本是由 GitHub 用户 fxsjy 开发的，根据 GitHub 个人信息显示，他工作于北京百度公司。安装此工具包之后，可以在 Python 中通过 import jieba，来引用其中一系列的功能。其特点有：

- 支持三种分词模式：
    - 精确模式，试图将句子最精确地切开，适合文本分析；
    - 全模式，把句子中所有的可以成词的词语都扫描出来，速度非常快，但是不能解决歧义；
    - 搜索引擎模式，在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。
- 支持繁体分词
- 支持自定义词典

2. 过程模型

由于这是一个较为轻量级的软件（或者更准确来说是组件），其过程模型更像是敏捷方法（Agile 方法）：根据用户的反馈，在短时间间隔内交付软件增量。其比较符合的基本过程模型是增量模型。

《大教堂和集市》一书被视为是开放源代码运动的《圣经》，这本书的作者 Eric Raymond 成为了领导开放源代码运动的理论家和开放源代码促进会（Open Source Initiative）的主要创办人之一。在书中，他认为世上的建筑可以分为两种：大教堂和集市。大教堂需要几代人呕心沥血，几十年才能建成辉煌巨制，投入使用，鲜少修改；集市则是天天开放，从无到有，从小到大。他用这两种建筑类比两种自由软件的开发模式：大教堂模式和集市模式。

大教堂模式指的是源代码在软件发行后公开，但是软件的每个版本开发过程中，是由一个专属的团队控管的；而市集模式是源代码在开发过程中即在互联网上公开，供所有人检视

和开发。GitHub 上的项目，实际上采用的是这两种方法的综合，此项目正是如此。

由于这是发布在 GitHub 上的开源项目，故其生存周期中的一系列相关活动是较为连续、迭代的。项目本身非商业性质，故初始版本的需求方未知，姑且可以认为项目需求方即为项目开发方。即程序员在其工作过程中遇到相应问题即需求，给自己布置项目并完成，然后开源提供给潜在的有同样需求的用户，并且在随后的过程中进行软件维护。

项目每次提供的都是可以直接执行的版本，但是由于项目一直处于更新状态中，故都不是最终版本，而是被视为是中间版本。每一项新的版本都是根据前一版本发布后收到的需求反馈，从用户最重视的需求出发，构造增加了更多功能或者解决了前一版本问题的新版本。

新版本的需求是通过用户反馈的需求进行确定的。每一个用户可以 fork 并 clone 源代码，即将源代码复制一遍。然后用户可在本地修改过后的版本中进行修改，并且将修改过后的版本通过 pull request 将修改后的代码同文档一起发送给开发者，即告诉开发者“我认为这些地方需要修订，我的修订方式如下”。由开发者选择是否接受这些修订意见，如果接收，网络上公开的软件版本就将发生更改。同样地，开发者也可以进行软件的修改。

主要的特点在于软件的修订开发内容，可以由软件开发者或者其他用户参与，但是最后由软件开发者在其个人平台上发布。自然，其他用户可在其个人平台上自由发布其修订后的版本。这既保证了原始发起开发者的决定权，也集中了更多的开发者的能力参与其中。

3. 改进情况

从 2012 年 10 月上线到 2015 年 12 月最近一次更新日志，结巴分词总共发布了 29 次更新信息（包括初次发布）。其中在 2013 年 7 月前的更新较为频繁，之后可能版本较为成熟，或者原始发起开发者的工作重心发生偏移，故更新信息较少。下面摘选了部分更新信息，可以看到前期主要是进行一些算法的开发实现调整，提升了工作的速度；后面增添了更多的功能；再后来主要是进行一些常见不常见的错误 bug 的调整。且后期的更新当中有更多的其他开发者的参与。

2012-10-07: version 0.14

==========

1) 结巴分词被发布到了 pypi，用户可以通过 easy_install 或者 pip 快速安装该组件；
2) 合并了搜狗开源词库 2006 版，删除了一些低频词
3) 优化了代码，缩短了程序初始化时间。
4) 增加了在线效果演示

2012-10-09: version 0.16

==========

1）将求最优切分路径的记忆化递归搜索算法改用循环实现，使分词速度提高了 15%
2) 修复了 Viterbi 算法实现上的一个 Bug

2012-12-28: version 0.24

==========

1) 解决了没有标点的长句子分词效果差的问题，问题在于连续的小概率乘法可能会导致浮点下溢或为 0.
2) 修复了 0.23 的全模式下英文分词的 bug

2013-04-07: version 0.26

==========

1) 改进了对标点符号的处理，之前的版本会过滤掉所有的标点符号;
2) 允许用户在自定义词典中添加词性;
3) 改进了关键词提取的功能 jieba.analyse.extract_tags;
4) 修复了一个在 pypy 解释器下运行的 bug.

2013-04-27: version 0.28

==========

1) 新增词典 lazy load 功能，用户可以在'import jieba'后再改变词典的路径. 感谢 hermanschaaf
2) 显示词典加载异常时错误的词条信息. 感谢 neuront
3) 修正了词典被 vim 编辑后会加载失败的 bug. 感谢 neuront

2013-07-01: version 0.30

==========

1) 新增 jieba.tokenize 方法，返回每个词的起始位置
2) 新增 ChineseAnalyzer，用于支持 whoosh 搜索引擎
3）添加了更多的中英混合词汇
4）修改了一些 py 文件的加载方法，从而支持 py2exe,cxfree 打包为 exe

2013-07-01: version 0.31

==========

1. 修改了代码缩进格式，遵循 PEP8 标准
2. 支持 Jython 解析器，感谢 @piaolingxue
3. 修复中英混合词汇不能识别数字在前词语的 Bug
4. 部分代码重构，感谢 @chao78787
5. 多进程并行分词模式下自动检测 CPU 个数设置合适的进程数，感谢@linkerlin
6. 修复了 0.3 版中 jieba.extra_tags 方法对 whoosh 模块的错误依赖

2014-08-31: version 0.33

==========

1. 支持自定义 stop words; by @fukuball
2. 支持自定义 idf 词典; by @fukuball
3. 修复自定义词典的词性不能正常显示的 bug; by @ShuraChow

2014-11-13: version 0.35

==========

1. 改进词典 cache 的 dump 和加载机制；by @gumblex
2. 提升关键词提取的性能; by @gumblex
3. 关键词提取新增基于 textrank 算法的子模块; by @singlee
4. 修复自定义 stopwords 功能的 bug; by @walkskyer

2015-03-20: version 0.36

==========

1. 代码同时兼容 python2 与 python3，若干性能优化; by @gumblex
2. 解决用户添加词的概率自动计算问题，分词更加准确；by @gumblex
3. 可自定义 cache_file 的文件系统路径; by @changyy
4. TextRank 算法实现完善; by @sing1ee，@walkskyer

2015-06-27: version 0.37

==========

1. 代码重构，分词器封装为 Class，支持实例化，by @gumblex (https://github.com/fxsjy/jieba/commit/94840a734c32cfece05c0c3ec236ffc3d36b4ae6)
2. 修复 cut_for_search 的 bug，完善 posseg； by @gumblex
3. 修复 posseg 在 0.36 中引入的一处 bug; by @wangbin
4. 修复 load_userdict 异常处理的 bug; by @gip0
5. 修复生成词典二进制 cache 文件时跨文件系统的 bug, 支持自定义; by @gumblex

2015-12-16: version 0.38

==========

1. 通过 pkg_resources 载入默认词典，支持在 Spark 等平台上运行, by @gumblex;
2. 扩充识别的汉字 unicode 范围：[\u4E00-\u9FD5], by @gumblex;
3. 关键词提取支持返回词性，修复 posseg 分词得到的 pair 做 dict 关键字的问题，by @jerryday；
4. 修复 load_userdict 加载用户词典不能识别含有空格等特殊字符的问题， by @gumblex;
5. 命令行分词支持返回词性， by @gumblex;

4. 总结

可以看到，GitHub 这样的在线版本管理平台，创建了一种新型的软件开发模式：使用者既是用户，又可以充当开发者。使用者自身对于需求的拿捏是更加准确的，他们如果有能力对于软件进行新一轮迭代的开发，这将为软件的开发汇聚更多的力量。管理者在享受众人智慧集合的同时，也需要进一步提升管理能力。另外，当管理者无精力继续此项目时，可能需要他人进行接手，成为新的核心开发者，其他人则在新的开发者的管理下继续使用、开发。