首页 开源项目 问答 代码 博客 翻译 资讯 移动开发 招聘 城市圈
当前访客身份：游客 [登录 | 加入开源中国]

樂天 关注此人
关注(10) 粉丝(100) 积分(225)
积累 && 远离舒适圈
.发送私信 .请教问题

博客分类

- 虚拟化(2)
- 中间件、KV(7)
- C/C++(21)
- Java(38)
- 面试、笔试、算法(43)
- 日常记录(122)
- PHP(3)
- J2EE(16)
- web前端(38)
- Android(21)
- 分布式、云(7)
- DB(36)
- web(20)
- 数据挖掘、数学、R语言(33)
- Node.js(6)
- latex(8)
- Python(49)
- 中文分词、全文检索(9)
- Linux、服务器、运维(84)
- Qt(15)
- vim/emacs(15)

阅读排行

1. 1. Python： pip常见的使用方法
2. 2. 使用TextRank算法为文本生成关键字和摘要
3. 3. IntelliJ IDEA 12 与 Tomcat7 配置
4. 4. Java下unicode转换为utf-8
5. 5. 整合Lucene 4.10.1 与IK Analyzer
6. 6. 解决sqlite3.DatabaseError: file is encrypted or ...
7. 7. 在Atom中运行脚本
8. 8. R：朴素贝叶斯

最新评论

- @okSola：引用来自“codekun”的评论发现用这个运行pytho... 查看»
- @okSola：你现在解决了吗？我也是这个问题，而且如果程序里... 查看»
- @樂天：引用来自“崔光范”的评论请问怎样转载这篇博客，... 查看»
- @崔光范：请问怎样转载这篇博客，转不到自己的空间里。。。 查看»
- @樂天：引用来自“萧雅祺”的评论再请教一下博主，每次执... 查看»
- @萧雅祺：再请教一下博主，每次执行都要 conn.commit()吗？... 查看»
- @萧雅祺：引用来自“萧雅祺”的评论from __future__ impor... 查看»
- @萧雅祺：引用来自“萧雅祺”的评论from __future__ impor... 查看»
- @萧雅祺：引用来自“萧雅祺”的评论from __future__ impor... 查看»
- @樂天：引用来自“貌似掉线”的评论String s1 = "\u5e7f... 查看»

友情链接

- 1. google本博客

原　荐

使用TextRank算法为文本生成关键字和摘要

发表于1年前(2014-12-01 21:31)　阅读（10215）| 评论（27）　155人收藏此文章, 我要收藏

赞15

摘要 TextRank算法基于PageRank，用于为文本生成关键字和摘要。

pagerank textrank 自动摘要 关键词 提取

目录[-]

TextRank算法基于PageRank，用于为文本生成关键字和摘要。其论文是：

> Mihalcea R, Tarau P. TextRank: Bringing order into texts[C]. Association for Computational Linguistics, 2004.

先从PageRank讲起。

PageRank

PageRank最开始用来计算网页的重要性。整个www可以看作一张有向图图，节点是网页。如果网页A存在到网页B的链接，那么有一条从网页A指向网页B的有向边。

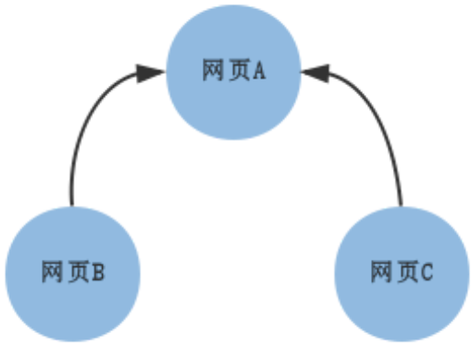
构造完图后，使用下面的公式：

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

S(Vi)是网页i的中重要性（PR值）。d是阻尼系数，一般设置为0.85。In(Vi)是存在指向网页i的链接的网页集合。Out(Vj)是网页j中的链接存在的链接指向的网页的集合。|Out(Vj)|是集合中元素的个数。

PageRank需要使用上面的公式多次迭代才能得到结果。初始时，可以设置每个网页的重要性为1。上面公式等号左边计算的结果是迭代后网页i的PR值，等号右边用到的PR值全是迭代前的。

举个例子：



上图表示了三张网页之间的链接关系，直觉上网页A最重要。可以得到下面的表：

结束\起始	A	B	C
A	0	1	1
B	0	0	0
C	0	0	0

横栏代表其实的节点，纵栏代表结束的节点。若两个节点间有链接关系，对应的值为1。

根据公式，需要将每一竖栏归一化（每个元素/元素之和），归一化的结果是：

结束\起始	A	B	C
A	0	1	1
B	0	0	0
C	0	0	0

上面的结果构成矩阵M。我们用matlab迭代100次看看最后每个网页的重要性：

```
M = [0 1 1
    0 0 0
    0 0 0];

PR = [1; 1 ; 1];

for iter = 1:100
    PR = 0.15 + 0.85*M*PR;
    disp(iter);
    disp(PR);
end
```

运行结果（省略部分）：

```
......

    95

    0.4050
    0.1500
    0.1500

    96

    0.4050
    0.1500
    0.1500

    97

    0.4050
    0.1500
```

```
    0.1500

    98

    0.4050
    0.1500
    0.1500

    99

    0.4050
    0.1500
    0.1500

   100

    0.4050
    0.1500
    0.1500
```

最终A的PR值为0.4050，B和C的PR值为0.1500。

如果把上面的有向边看作无向的（其实就是双向的），那么：

```
M = [0 1 1
    0.5 0 0
    0.5 0 0];

PR = [1; 1 ; 1];

for iter = 1:100
    PR = 0.15 + 0.85*M*PR;
    disp(iter);
    disp(PR);
end
```

运行结果（省略部分）：

```
.....

    98

    1.4595
    0.7703
    0.7703

    99

    1.4595
    0.7703
    0.7703

   100

    1.4595
    0.7703
    0.7703
```

依然能判断出A、B、C的重要性。

使用TextRank提取关键字

将原文本拆分为句子，在每个句子中过滤掉停用词（可选），并只保留指定词性的单词（可选）。由此可以得到句子的集合和单词的集合。

每个单词作为pagerank中的一个节点。设定窗口大小为k，假设一个句子依次由下面的单词组成：

```
w1, w2, w3, w4, w5, ..., wn
```

w1, w2, ..., wk、w2, w3, ...,wk+1、w3, w4, ...,wk+2等都是一个窗口。在一个窗口中的任两个单词对应的节点之间存在一个无向无权的边。

基于上面构成图，可以计算出每个单词节点的重要性。最重要的若干单词可以作为关键词。

使用TextRank提取关键短语

参照“使用TextRank提取关键词”提取出若干关键词。若原文本中存在若干个关键词相邻的情况，那么这些关键词可以构成一个关键短语。

例如，在一篇介绍“支持向量机”的文章中，可以找到三个关键词支持、向量、机，通过关键短语提取，可以得到支持向量机。

使用TextRank提取摘要

将每个句子看成图中的一个节点，若两个句子之间有相似性，认为对应的两个节点之间有一个无向有权边，权值是相似度。

通过pagerank算法计算得到的重要性最高的若干句子可以当作摘要。

论文中使用下面的公式计算两个句子Si和Sj的相似度：

$$Similarity(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{log(|S_i|) + log(|S_j|)}$$

分子是在两个句子中都出现的单词的数量。|Si|是句子i的单词数。

由于是有权图，PageRank公式略做修改：

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j)$$

实现TextRank

因为要用测试多种情况，所以自己实现了一个基于Python 2.7的TextRank针对**中文文本**的库TextRank4ZH。位于：

https://github.com/someus/TextRank4ZH

下面是一个例子：

```python
#-*- encoding:utf-8 -*-

import codecs
from textrank4zh import TextRank4Keyword, TextRank4Sentence

text = codecs.open('./text/01.txt', 'r', 'utf-8').read()
tr4w = TextRank4Keyword(stop_words_file='./stopword.data')  # 导入停止词

#使用词性过滤，文本小写，窗口为2
tr4w.train(text=text, speech_tag_filter=True, lower=True, window=2)
```

```
11
12  print '关键词：'
13  # 20个关键词且每个的长度最小为1
14  print '/'.join(tr4w.get_keywords(20, word_min_len=1))
15
16  print '关键短语：'
17  # 20个关键词去构造短语，短语在原文本中出现次数最少为2
18  print '/'.join(tr4w.get_keyphrases(keywords_num=20, min_occur_num= 2))
19
20  tr4s = TextRank4Sentence(stop_words_file='./stopword.data')
21
22  # 使用词性过滤，文本小写，使用words_all_filters生成句子之间的相似性
23  tr4s.train(text=text, speech_tag_filter=True, lower=True, source = 'all_fil
24
25  print '摘要：'
26  print '\n'.join(tr4s.get_key_sentences(num=3)) # 重要性最高的三个句子
```

运行结果如下：

```
1  关键词：
2  媒体/高圆圆/微/宾客/赵又廷/答谢/谢娜/现身/记者/新人/北京/博/展示/捧场/礼物/张杰/当
3  关键短语：
4  微博
5  摘要：
6  中新网北京12月1日电(记者 张曦) 30日晚，高圆圆和赵又廷在京举行答谢宴，诸多明星现身捧
7  高圆圆身穿粉色外套，看到大批记者在场露出娇羞神色，赵又廷则戴着鸭舌帽，十分淡定，两人
8  记者了解到，出席高圆圆、赵又廷答谢宴的宾客近百人，其中不少都是女方的高中同学
```

另外， jieba分词提供的基于TextRank的关键词提取工具。 snownlp也实现了关键词提取和摘要生成。

- « 上一篇
- 下一篇 »

评论27

- 
  1楼：封心 发表于 2014-12-03 08:58 回复此评论
  高大上的感觉

- 
2楼：吐槽的达达仔 发表于 2014-12-03 09:09 回复此评论
感觉像是名词动词提取的样子。。
- 
3楼：土豆哥哥好 发表于 2014-12-03 09:30 回复此评论
收下慢慢看
- 4楼：樂天 发表于 2014-12-03 09:48 回复此评论


引用来自“吐槽的达达仔”的评论

感觉像是名词动词提取的样子。。
词性过滤后，的确是名词动词居多
- 
5楼：xuehao822 发表于 2014-12-03 10:16 回复此评论
使用TextRank提取关键短语
介绍的很模糊啊
- 
6楼：Caskia 发表于 2014-12-03 10:30 回复此评论
为什么不用TF IDF呢
- 7楼：樂天 发表于 2014-12-03 10:39 回复此评论

引用来自“xuehao822”的评论


使用TextRank提取关键短语
介绍的很模糊啊
我觉得说的还行吧。假设关键短语只有两个单词组成，我们提取的关键词中有“你好”“世界”这两个词。如果”你好世界“这个短语出现在文本中若干次，则认为”你好世界“是一个关键短语。不知道这样的解释行不行？
- 8楼：樂天 发表于 2014-12-03 10:40 回复此评论


引用来自“Caskia”的评论

为什么不用TF IDF呢
TF-IDF是一个思路。我记得阮一峰老师有过介绍。
- 
9楼：火烧 发表于 2014-12-03 11:06 回复此评论
高大上
- 
10楼：测试小松鼠 发表于 2014-12-03 11:16 回复此评论
流弊到理解不了

- 1
- 2
- 3
- >

插入： 表情 开源软件

发表评论

关闭插入表情

关闭相关文章阅读

- 2015/07/20 TextRank算法及用途
- 2014/01/19 PageRank算法
- 2013/03/19 pagerank-mapreduce
- 2013/04/22 自动摘要
- 2015/05/11 hadoop下基于mapreduce实现pageran...