

SnowNLP 代码阅读报告

一、SnowNLP 定义

一个处理中文文本的 Python 类库，没有用 NLTK，所有的算法都是自己实现的，并且自带了一些训练好的字典。

二、SnowNLP 用途

- 中文分词 (Character-Based Generative Model)
s.words
- 词性标准
s.tags
- 情感分析 (目前主要针对买卖东西时的评价)
 - ✓ 返回值为正面情绪的概率
 - ✓ 越接近 1 表示正面情绪
 - ✓ 越接近 0 表示负面情绪**s.sentiments**
- 文本分类 (Naive Bayes)
- 转换成拼音 (Trie 树实现的最大匹配)
s.pinyin
- 繁体转简体 (Trie 树实现的最大匹配)
s.han
- 提取文本关键词 (TextRank 算法)
s.keywords(number)
- 提取文本摘要 (TextRank 算法)
s.summary(number)
- 信息量衡量 TF-IDF
 - ✓ TF-IDF 是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。
 - ✓ TF 词频越大越重要，但是文中会的“的”，“你”等无意义词频很大，却信息量几乎为 0，这种情况导致单纯看词频评价词语重要性是不准确的。因此加入了 idf。
 - ✓ IDF 的主要思想是：如果包含词条 t 的文档越少，也就是 n 越小，IDF 越大，则说明词条 t 越重要
 - ✓ TF-IDF 综合起来，才能准确的综合的评价一词对文本的重要性。
- Tokenization (分割成句子)
- 文本相似度计算 (BM25)
s.sim

□ 名称

classification

normal

seg


sentiment

sim

summary

tag

utils

 __init__.py

Normal – 转换成拼音、简繁转换、分句

Seg – 中文分词

Sentiment – 情感分析

Sim – 文本相似度

Summary – 提取文本关键词, 提取文本摘要


Tag – 词性标注


Utils – 其他辅助函数


```
1 # -*- coding: utf-8 -*-
2 from __future__ import unicode_literals
3
4 from . import normal
5 from . import seg
6 from . import tag
7 from . import sentiment
8 from .sim import bm25
9 from .summary import textrank
10 from .summary import words_merge
11
12
13 class SnowNLP(object):
14
15     def __init__(self, doc):
16         self.doc = doc
17         self.bm25 = bm25.BM25(doc)
18
19     @property
20     def words(self):
21         return seg.seg(self.doc)
22
23     @property
24     def sentences(self):
25         return normal.get_sentences(self.doc)
26
27     @property
28     def han(self):
29         return normal.zh2hans(self.doc)
```


__init__() 一般用于初始化一个类


Normal

 __init__.py

 pinyin.py

 pinyin.txt

 stopwords.txt

 zh.py

拼音汉字对应文本 去停用词文本

zh.py 简繁转换

Seg

名称	修改日期	类型
init.py	2017/5/19 1:50	Python File
data.txt	2017/5/19 1:50	文本文档
seg.marshall	2017/5/19 1:50	MARSHAL 文件
seg.marshall.3	2017/5/19 1:50	3 文件
seg.py	2017/5/19 1:50	Python File
y09_2047.py	2017/5/19 1:50	Python File

data.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

迈/b 向/e 充/b 满/e 希/b 望/e 的/s 新/s 世/b 纪/e 一/b 一/e 一/b 九/m 九/m 八/m 年/e 新/b 年/e 讲/b 话/e (/s 附/s 图/b 片/e 1/s 张/s) /s中/b 共/m 中/m 央/e 总/b 书/m 记/e 、 /s 国/b 家/e 主/b 席/e 江/s 泽/b 民/e (/s 一/b 九/m 九/m 七/m 年/e 十/b 二/m 月/e 三/b 十/m 一/m 日/e) /s 1/b 2/m 月/e 3/b 1/m 日/e , /s 中/b 共/m 中/m 央/e 总/b 书/m 记/e 、 /s 国/b 家/e 主/b 席/e 江/s 泽/b 民/e 发/b 表/e 1/b 9/m 9/m 8/m 年/e 新/b 年/e 讲/b 话/e 《/s 迈/b 向/e 充/b 满/e 希/b 望/e 的/s 新/s 世/b 纪/e 》/s 。 /s (/s 新/b 华/m 社/e 记/b 者/e 兰/s 红/b 光/e 摄/s) /s同/b 胞/e 们/s 、 /s 朋/b 友/e 们/s 、 /s 女/b 士/e 们/s 、 /s 先/b 生/e 们/s : /s在/s 1/b 9/m 9/m 8/m 年/e 来/b 临/e 之/b 际/e , /s 我/s 十/b 分/e 高/b 兴/e 地/s 通/b 过/e 中/b 央/e 人/b 民/e 广/b 播/e 电/b 台/e 、 /s 中/b 国/e 国/b 际/e 广/b 播/e 电/b 台/e 和/s 中/b 央/e 电/b 视/m 台/e , /s 向/s 全/b 国/e 各/b 族/e 人/b 民/e , /s 向/s 香/b 港/e 特/b 别/e 行/b 政/m 区/e 同/b 胞/e 、 /s 澳/b 门/e 和/s 台/b 湾/e 同/b 胞/e 、 /s 海/b 外/e 侨/b 胞/e , /s 向/s 世/b 界/e 各/b 国/e 的/s 朋/b 友/e 们/s , /s 致/b 以/e 诚/b 挚/e 的/s 问/b 候/e 和/s 良/b 好/e 的/s 祝/b 愿/e ! /s 1/b 9 /m 9/m 7/m 年/e , /s 是/s 中/b 国/e 发/b 展/e 历/b 史/e 上/s 非/b 常/e 重/b 要/e 的/s 很/s 不/s 平/b 凡/e 的/s 一/s 年/s 。 /s 中/b 国/e 人/b 民/e 决/b 心/e 继/b 承/e 邓/s 小/b 平

利用 data 进行分词训练

Sentiment

- _init_.py
- neg.txt
- pos.txt
- sentiment.marshall
- sentiment.marshall.3

存储正负情感色彩的词进行情感分析训练

Sim

- _init_.py
- bm25.py

BM25 用来计算 query 和文章相关度的相似度

算法的原理：将需要计算的 query 分词成 w_1, w_2, \dots, w_n ，然后求出每

一个词和文章的相关度，最后将这些相关度进行累加，最终就可以得到文本相似度计算结果。

Summary

Textrank 提取关键词短语或摘要

<https://www.cnblogs.com/bymo/p/8462120.html>

tag

利用文本进行词性标注的训练

Trie 树

即前缀树，字典树，单词查找树或键树。

典型应用是用于统计和排序大量的字符串（但不仅限于字符串），所以经常被搜索引擎系统用于文本词频统计。

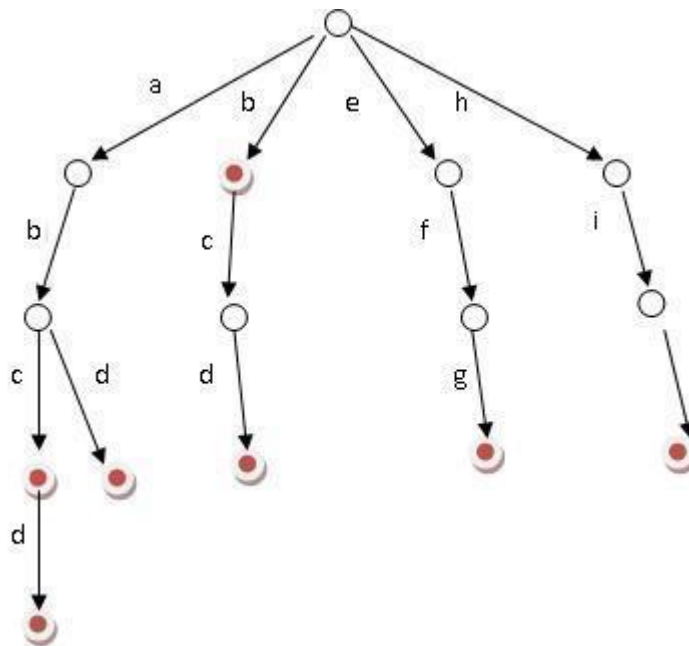
它的优点是：最大限度地减少无谓的字符串比较。

Trie 的核心思想是空间换时间。利用字符串的公共前缀来降低查询时间的开销以达到提高效率的目的。

3 个基本性质：

- 根节点不包含字符，除根节点外每一个节点都只包含一个字符。
- 从根节点到某一节点，路径上经过的字符连接起来，为该节点对应的字符串。
- 每个节点的所有子节点包含的字符都不相同。

假设有 b, abc, abd, bcd, abcd, efg, hii 这 6 个单词，我们构建的树就是如下图这样的：



<https://github.com/chexiangcyr/leetcode-answers>

<https://www.jianshu.com/p/a57c44a74bf9>

re 模块提供正则表达式匹配操作

`re.compile(pattern[, flags])` 根据包含正则表达式的字符串创建模式对象，`flags` 是匹配模式，可以实现更有效的匹配。

`Re.match(pattern, string, flags = 0)` 如果在字符串的开头的零个或者更多字符匹配正则表达式模式，将返回相应的 `MatchObject` 实例。

<https://www.cnblogs.com/tina-python/p/5508402.html>