文本相似度

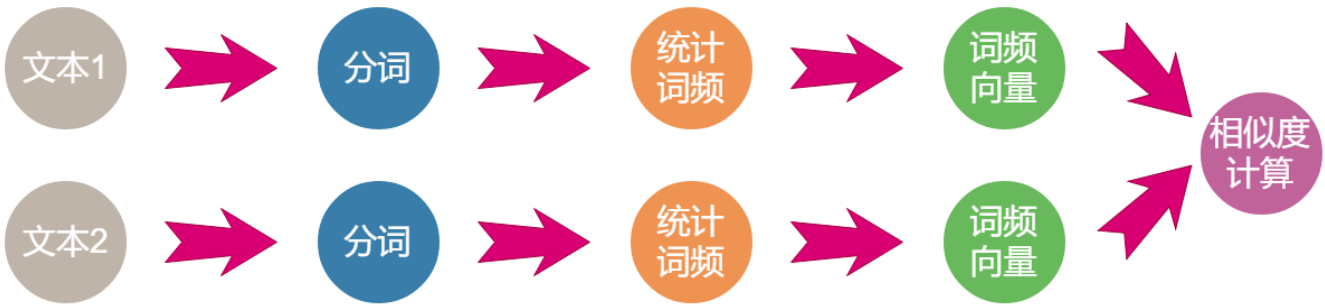
应用

- 论文检测
  - 知网论文检测 http://check.cnki.net/
  - Paper Pass论文检测http://www.paperpass.com/
- 其它领域研究的基础：文本聚类，文本分类 ，文本挖掘，信息检索，舆情分析

原理

- 基于词频：统计词频，构建词频特征向量，利用特征向量夹角余弦表示文本相似度。
- 基于语义：基于语料统计的方法，基于word-Net, How-Net词典提取相关语义特征。

基于词频的文本相似度



统计文本中每个词出现的次数，即词频，通过词频构建文本向量，通过计算两个文本向量之间的余弦相似度，反映两个文本之间的相似度。

- 文本分词
- 去停用词
- 统计词频，构建词频向量
- 词频向量余弦相似度

中文分词

词是最小的能够独立活动的有意义的语言成分，英文单词之间是以空格作为自然分界符的，而汉语是以字为基本的书写单位，词语之间没有明显的区分标记，因此中文是一定要分词的。而且nlp的基础任务中，关键词抽取，词性标注，命名实体识别，语法分析，句法分析等等都默认了词是基本单位。

中文分词原理：https://www.cnblogs.com/BaiYiShaoNian/p/5071802.html

- Jieba (C++, Java, python)https://github.com/fxsjy/jieba
- HanLP (Java)https://github.com/hankcs/HanLP
- FudanNLP (Java)https://github.com/FudanNLP/fnlp
- LTP (C++, Java, python)https://github.com/HIT-SCIR/ltp

举个栗子：

- 英文：I am a student-----> I // am // a // student 中文：我是一个学生 ---> 我 // 是 // 一个 // 学生
- 英文：Xi'an China
- 中文：中国西安

停用词

人类语言包含很多功能词。与其他词相比，功能词没有什么实际含义。 停用词主要包括数字、标点符号及使用频率特高的词(代词，语气助词、副词、介词、连接词)等。

我

我们

怎么办

总之

此外 然而 不如 不妨 。 ，？

.....

停用词不代表实际意义，所以不需要统计停用词的词频，停用词不参与构建词频向量

词频

- 词频即为单词在文章中出现的次数。
- 词频的大小一般可以反映一个词在一篇文章中的重要性，词频越大，可以认为该词越重要。
- 一片文章的语义可以由一组关键词简要概括，比如"今天早上八点钟，我要去比特上课"，关键词"八点，比特，上课"。

分词编码

在构建文本词频向量时，需要考虑向量的意义，也必须保证向量的一致性，这样才有可比性。

- 意义：文本的语义，用词频来表示
- 一致性：如何保证一致性？向量中的每一维值都应该表示相同的意思。

更具体的说，一致性就体现在两个文本向量的每一维都应该表示同一个词的词频。

举个栗子：

文档1：今天/有事/，/没办法/去/比特/上课/了

文档2：真想/去/比特/上课/，/但是/今天/有事/，/去不了/比特/了

文档1中的词频：[今天：1，有事：1，没办法：1，去：1，比特：1，上课：1，了：1]

文档2中的词频：[真想：1，去：1，比特：2，上课：1，但是：1， 今天：1，有事：1，去不了：1，了：1]

去掉停用词之后：

文档1中的词频：[有事：1，没办法：1，去：1，比特：1，上课：1]

文档2中的词频：[真想：1，去：1，比特：2，上课：1， 有事：1，去不了：1]

直接用上述词频构建每一个文本的词频向量无意义，每一维表示的意思不同，两个向量没有可比性。

构建一致的词频向量：给每一维的词频编码，然后去看每一维的词频向量。

- 把两个文本中的所有有效词全部编码，对于长文本可以按词频从大到小排序，取前n个关键词
- 按照码值构建词频向量

比如：

文档1中的词频：[有事：1，没办法：1，去：1，比特：1，上课：1]

文档2中的词频：[真想：1，去：1，比特：2，上课：1， 有事：1，去不了：1]

所有有效词：比特，去，真想，上课，有事，去不了，没办法

给所有有效词编码：比特：0，去：1，真想：2，上课：3，有事：4，去不了：5，没办法：6

词频向量

通过上述词的编码值，构建词频向量

文档1中的词频：[0：1，1：1，2：0，3：1，4：1，5：0，6：1]

文档2中的词频：[0：2，1：1，2：1，3：1， 4：1，5：1，6：0]

文档1词频向量：[1，1，0，1，1，0，1]

文档2词频向量：[2，1，1，1， 1，1，0]

向量相似度

常用计算向量相似度的方式：欧几里得距离，余弦相似度，jaccard系数(类似余弦相似度)，曼哈顿距离(类似欧几里得距离)。

余弦相似度，是通过计算两个向量的夹角余弦值来评估他们的相似度。

$$cos(\theta) = \frac{A \bullet B}{||A|| \bullet ||B||} = \frac{\sum_i^n a_i * b_i}{\sqrt{\sum_i^n a_i^2} * \sqrt{\sum_i^n b_i^2}}$$

附录

windows编码转换接口

- MultiByteToWideCharhttps://docs.microsoft.com/en-us/windows/desktop/api/stringapiset/nf-stringapiset-multibytetowidechar

```
/*
    Maps a character string to a UTF-16 (wide character) string.
*/
int MultiByteToWideChar(
  UINT                              CodePage,
  DWORD                             dwFlags,
  _In_NLS_string_(cbMultiByte)LPCCH lpMultiByteStr,
  int                               cbMultiByte,
  LPWSTR                            lpWideCharStr,
  int                               cchWideChar
);
```

- [ ] CodePage：执行转换的字符编码，即要转换字符的编码格式。(CP_ACP(windows系统默认编码格式)，CP_MACCP(MacOS)，CP_UTF8等)。
- [ ] dwFlags：转换类型标记，这里只需注意对于UTF8或者GBK，此值要设为0或者 MB_ERR_INVALID_CHARS(转换失败)。
- [ ] lpMultiByteStr：要转换的字符串指针。
- [ ] cbMultiByte：要转换的字节大小，如果此值设为-1，则处理整个字符串，包括结束字符(比如'\0')，函数返回的字符串长度也包括结束字符。
- [ ] lpWideCharStr：保存转换之后的字符串buffer(UTF-16)。
- [ ] cchWideChar：lpWideCharStr的buffer大小。如果此值为0，函数返回buffer所要求的大小，包括任何结束字符。
- [ ] 返回值：写入到lpWideCharStr的buffer字符数量。

- WideCharToMultiBytehttps://docs.microsoft.com/en-us/windows/desktop/api/stringapiset/nf-stringapiset-widechartomultibyte

```
/*
    Maps a UTF-16 (wide character) string to a new character string.
*/
int WideCharToMultiByte(
  UINT                               CodePage,
  DWORD                              dwFlags,
  _In_NLS_string_(cchWideChar)LPCWCH lpWideCharStr,
  int                                cchWideChar,
  LPSTR                              lpMultiByteStr,
  int                                cbMultiByte,
  LPCCH                              lpDefaultChar,
  LPBOOL                             lpUsedDefaultChar
);
```

  - [ ] CodePage：执行转换的编码格式，即要把UTF-16格式的字符转换成指定的编码格式。
  - [ ] dwFlags：转换类型标记，这里只需注意对于UTF8或者GBK，此值要设为0或者 WC_ERR_INVALID_CHARS(转换失败)。
  - [ ] lpWideCharStr：要转换的UTF-16字符串指针。
  - [ ] cchWideChar：要转换的UTF-16字符数量，如果此值设为-1，则处理整个字符串，包括结束字符(比如'\0')，函数返回的字符串长度也包括结束字符。
  - [ ] lpMultiByteStr：接收转换之后字符串的buffer指针。
  - [ ] cbMultiByte：buffer字节大小，如果此值为0，函数返回buffer所要求的大小。
  - [ ] 返回值：返回写入到buffer中的字节数。
  - [ ] lpDefaultChar，lpUsedDefaultChar：，默认检查，一般设为NULL。

拓展

- 用其它度量方式计算相似度