

文本数据分析及挖掘

自然语言处理

自然语言处理（natural language processing, NLP）

- 探讨如何处理及运用自然语言
 - 任务：
 - 词量统计，词频统计
 - 识别句子边界
 - 词性标注（part-of-speech tagging, POS tagging）
 - 解析句子结构
 - 语义角色标注
 - ...
-

中文分词技术

- 英文分词 vs 中文分词
 - 英文单词之间是以空格作为自然分界符的
 - 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
- 比如：
 - 输入：他是研究生物化学的
 - 可能的分词：
 - 他是研究生 物化 学的
 - 他是研究生 物 化学 的
 - 他是研究 生物 化学 的



分词工具

分词 (tokenize)

- 将句子拆分成具有语言语义学上意义的词
 - 中英文分词区别
 - 英文中，单词之间是以空格作为自然分界符的
 - 中文中没有一个形式上的分界符，分词比英文复杂的多
 - 英文分词工具，NLTK `pip install nltk`
 - 中文分词工具，如结巴分词 `pip install jieba`
 - 得到分词结果后，中英文的后续处理没有太大区别
-

中文分词工具

结巴分词

- 分词模式
 1. 精确模式：对语句进行最精确地切分
 2. 全模式：把句子中所有的可以成词的词语都扫描出来，但是不能解决歧义
 3. 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。
 - `jieba.cut(sentence, cut_all=False, HMM=True)`
 - `cut_all`:分词模式，`True` 代表全模式，`False` 代表精确模式，默认缺失值为`False`
 - 参考：<https://github.com/fxsjy/jieba>
-

分词相关操作

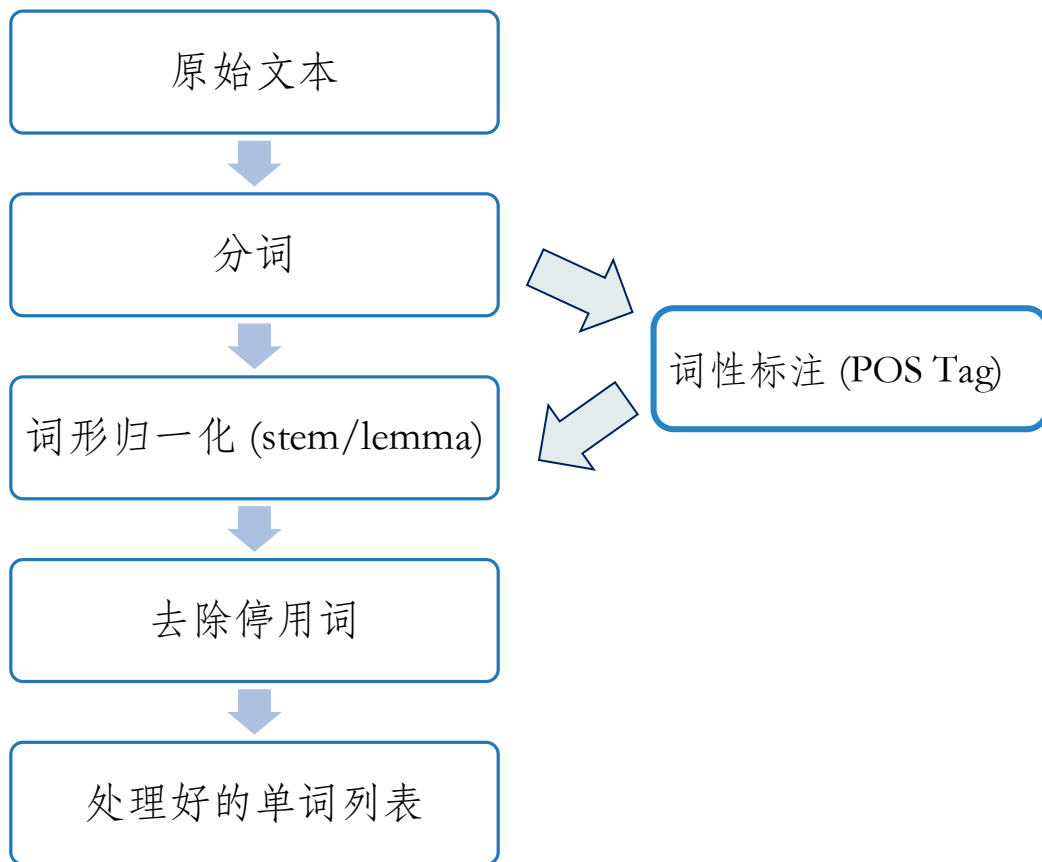
正则表达式去除特殊字符，如：标点符号...

停用词(Stopwords)

- 为节省存储空间和提高搜索效率，NLP中会自动过滤掉某些字或词
- 停用词都是人工输入、非自动化生成的，形成停用词表
- 分类
 - 语言中的功能词，如the, is...
 - 词汇词，通常是使用广泛的词，如“这”，“那” ...
 - 语气词，如“嗯”，“啊” ...
- 中文停用词表
 - 中文停用词库，哈工大停用词表，四川大学机器智能实验室停用词库
- 其他语言停用词表
 - <http://www.ranks.nl/stopwords>

文本预处理

典型的文本预处理流程



词袋模型

词袋模型包括：

1. 词典的构建
2. 如何度量词的出现

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

文本特征

- 使用词频表示文本特征
 - 文本中单词出现的频率或次数
 - 如

ID	nova	galaxy	heat	h'wood	film	role	diet	fur
A	10	5	3					
B	5	10						
C				10	8	7		
D				9	10	5		
E							10	10
F							9	10
G	5		7			9		
H		6	10	2	8			
I				7	5		1	3

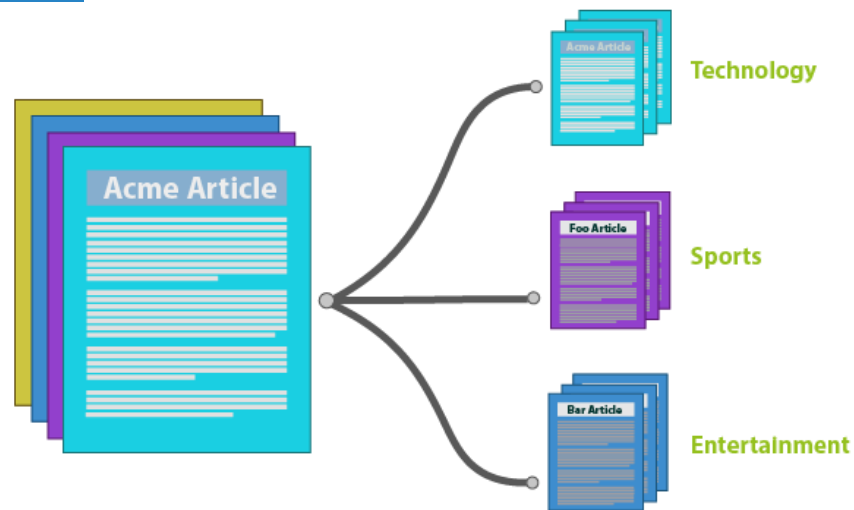
- 将文本表示成向量

`sklearn.feature_extraction.text.CountVectorizer`

TF-IDF

TF-IDF（词频-逆文档频率）

- TF, Term Frequency（词频）。某个词在该文件中出现的次数
- IDF, Inverse Document Frequency（逆文档频率），用于衡量某个词普遍的重要性。
- $TF\text{-}IDF = TF * IDF$



$$TF = \frac{\text{当前词在该文档中出现的次数}}{\text{文档中词的总数}}$$

$$IDF = \log\left(\frac{\text{总文档个数}}{\text{当前词出现的文档个数}}\right)$$

TF-IDF

TF-IDF（词频-逆文档频率）

- 例子
- 一个包含100个单词的文档中出现单词cat的次数为3，则 $TF=3/100=0.03$
- 样本中一共有10,000,000个文档，其中出现cat的文档数为1,000个，则 $IDF=\log(10,000,000/1,000)=4$
- $TF-IDF = TF * IDF = 0.03 * 4 = 0.12$
- IDF的其他计算形式： $idf(t) = \log \frac{1+n_d}{1+df(d,t)} + 1$
然后对tf-idf特征再做归一化 $v_{norm} = \frac{v}{||v||_2} = \frac{v}{\sqrt{v_1^2+v_2^2+\dots+v_n^2}}$
- 常用的文本特征：

http://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction
