### NLP基础

分词 jieba/nshort

NER

POS

情感分析 snownlp

句法依存

PPL srilm

文本相似度

文本分类 cnn/LSTM/RNN

文本聚类 kmeans knn

文本生成 seq2seq

词向量 word2vec

LDA gensim

TF/IDF

提取Topic/场景/关键词/Tag

几种常用计算相似度的算法：

### **杰卡德(Jaccard)相似系数**:计算单词集合之间的交集和并集大小的比例

****

**编辑距离:**两个字串之间，由一个转成另一个所需的最少编辑操作次数

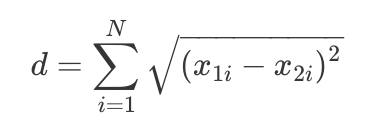
simhash + 汉明距离:汉明距离是使用在数据传输差错控制编码里面的，汉明距离是一个概念，它表示两个（相同长度）字对应位不同的数量，我们以d（x,y）表示两个字x,y之间的汉明距离。对两个字符串进行异或运算，并统计结果为1的个数，那么这个数就是汉明距离.在一个码组集合中，任意两个码字之间对应位上码元取值不同的位的数目定义为这两个码字之间的汉明距离。即

d(x,y)=∑x[i]⊕y[i]

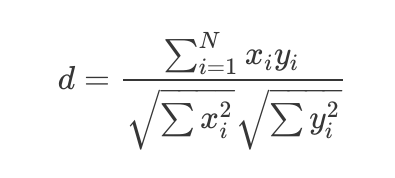
i=0,1,..n-1，x,y都是n位的编码，⊕表示异或

（据说:SimHash对于短文本误判率比较高，因此建议大于500字以上的使用此算法）

欧式距离:欧式距离源自N维欧氏空间中两点x1,x2间的距离公式



余弦相似度:计算两个向量之间的夹角，夹角越小相似度越高



|  |
| --- |
| 欧氏距离衡量的是空间各点的绝对距离，跟各个点所在的位置坐标直接相关；而余弦距离衡量的是空间向量的夹角，更加体现在方向上的差异，而不是位置  欧氏距离能够体现个体数值特征的绝对差异，所以更多的用于需要从维度的数值大小中体现差异的分析，如使用用户行为指标分析用户价值的相似度或差异。  余弦距离更多的是从方向上区分差异，而对绝对的数值不敏感，更多的用于使用用户对内容评分来区分兴趣的相似度和差异，同时修正了用户间可能存在的度量标准不统一的问题（因为余弦距离对绝对数值不敏感）。  正因为余弦相似度在数值上的不敏感，会导致这样一种情况存在：  用户对内容评分，按5分制，X和Y两个用户对两个内容的评分分别为（1,2）和（4,5），使用余弦相似度得到的结果是0.98，两者极为相似。但从评分上看X似乎不喜欢2这个 内容，而Y则比较喜欢，余弦相似度对数值的不敏感导致了结果的误差，需要修正这种不合理性就出现了调整余弦相似度，即所有维度上的数值都减去一个均值，比如X和Y的评分均值都是3，那么调整后为（-2，-1）和（1,2），再用余弦相似度计算，得到-0.8，相似度为负值并且差异不小，但显然更加符合现实。  <https://blog.csdn.net/zz_dd_yy/article/details/51926305> |

1.计算句子向量：先分词计算每个词的向量，然后把所有词的向量相加，得到句子的向量。

2.gensim

提取Topic/场景/关键词/Tag

TF/IDF：

缺点：

1. 更倾向于高频词。

（<http://hejunhao.me/archives/918>）

TextRank：

缺点：

1. 综合考虑文本结构和词频，但它致命的问题在于，头尾的信息由于只有单边的入度，容易被抛弃掉

LDA:

缺点：

1. LDA产生的主题往往被高频词占据，这种现象导致低频词在实际应用中的作用非常有限。 （<https://github.com/baidu/Familia/wiki>）

2. 短文本上效果很差？

TWE(Topic Word Embedding)

利用LDA训练获得的主题为词向量的训练提供补充信息，进而得到词和主题的向量表示。有鉴于向量表示可以较好地建模低频词的语义信息，通过利用词和主题的向量表示，我们可以更好地捕捉每个主题下的低频词的语义信息，提升下游应用的效果。