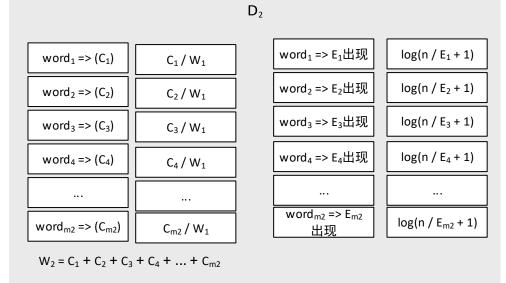


总文档数n 原始的TFIDF



•••••

		D_n			
word ₁ => (C ₁)	C ₁ / W ₁		word ₁ => E ₁ 出		
$word_2 \Rightarrow (C_2)$	C ₂ / W ₁		word ₂ => E ₂ 出		
$word_3 \Rightarrow (C_3)$	C ₃ / W ₁		word ₃ => E ₃ 出		
$word_4 \Rightarrow (C_4)$	C ₄ / W ₁		word ₄ => E ₄ 出		
word _{mn} => (C _{mn})	C _{mn} / W ₁		word _{mn} => E _r 出现		
$W_n = C_1 + C_2 + C_3 + C_4 + + C_{mn}$					

word ₁ => E ₁ 出现	log(n / E ₁ + 1)
word ₂ => E ₂ 出现	log(n / E ₂ + 1)
word ₃ => E ₃ 出现	log(n / E ₃ + 1)
word ₄ => E ₄ 出现	log(n / E ₄ + 1)
word _{mn} => E _{mn} 出现	log(n / E _{mn} + 1)

tf-idf = TF * IDF

@author: natasha_yang

@e-mail: ityangna0402@163.com

我的想法一:

比如问题(比较短的文本)集合 Q_1 的label是 A_1 可否将 Q_1 中所有的问题组成一个doc求TF-IDF的值那如果要预测新的问题的时候要怎么办??

我的想法二:

可以将 Q_1 中所有的问题组成一个doc求TF-IDF的值然后形成文档的词TF-IDF value的向量(词典顺序)求主题的相似性

我的想法三:

在求IDF的时候可否考虑文档X中的word在非X的文档中出现的次数(感觉传统的IDF不能很好的体现主题的无关性)比如说存在10个主题的文档,(机器)在类型1中出现了10000次,在其他9个类型文档中各出现一次(那么IDF=1)

我的想法四:

如果加上某个单词在当前X文档出现的次数/某个单词在所有文档中出现的次数作为权重,这个值越接近1/n就越不重要,但是这样的话就要注意输入集合的数据要平衡