



Contents n t U[-]

- 5.3.1 用Pandas库实现
- 5.3.2 用sklearn库实现
- 5.4 从词袋模型到N-gram
- ▼ 5.5 文档信息的分布式表示
 - 5.5.1 什么是分布式表示
 - 5.5.2 共现矩阵
 - 5.5.3 NNLM模型
 - 5.5.4 CBOW模型
- 5.6 实战：生成词向量
- ▼ 6 关键词提取
 - 6.1 关键词提取的基本思路
 - 6.2 TF-IDF 算法
 - ▼ 6.3 TF-IDF的具体实现
 - 6.3.1 jieba
 - 6.3.2 sklearn
 - 6.3.3 gensim
 - 6.4 TextRank算法
 - 6.5 实战练习
- ▼ 7 抽取文档主题
 - 7.1 主题模型的基本概念
 - 7.2 sklearn实现
 - 7.3 gensim实现
 - 7.4 结果的图形化呈现
 - 7.5 实战练习

7.5 实战练习

在其余参数全部固定不变的情况下，尝试分别用清理前矩阵、清理后原始矩阵、TF-IDF矩阵进行LDA模型拟合，比较分析结果。

在gensim拟合LDA时，分别将passes参数设置为1、5、10、50、100等，观察结果变化的情况，思考如何对该参数做最优设定。

请尝试对模型进行优化，得到对本案例较好的分析结果。

提示：使用gensim进行拟合更容易一些。

8 文档相似度

8.1 基本概念

8.2 词条相似度：word2vec

词袋模型不考虑词条之间的相关性，因此无法用于计算词条相似度。

分布式表达会考虑词条的上下文关联，因此能够提取出词条上下文中的相关性信息，而词条之间的相似度就可以直接利用此类信息加以计算。

目前主要使用gensim实现相应的算法。