基于TF-IDF的新闻检索系统

内容

- 数据集简单介绍
- 项目效果
- 具体实现步骤与所用工具
- 值得改进的地方

数据集简单介绍

- CNN/Daily Mail dataset https://cs.nyu.edu/~kcho/DMQA/

文档总数	平均句子数	平均长度
286,817	29.74	766

数据集简单介绍

Washington (CNN) -- Many Americans are planning to get a glimpse of Prince William and his wife, Kate Middleton when they arrive in New York City next week -- and it seems that President Barack Obama wants in as well.

The President has invited Prince William to meet with him in the Oval Office on Monday. Before The Duke of Cambridge leaves the White House grounds, Vice President Joe Biden and Dr. Biden are expected to host him in a separate event.

Kate will not join her husband on the D.C. leg of their three-day trip, opting to tour a child development center with New York City's first lady Chirlane McCray instead.

While it is unknown exactly what the two global leaders will discuss, Obama acknowledged William's commitment to raising awareness against illegal wildlife trafficking, an issue he is scheduled to talk about at the World Bank.

"The President welcomes the Prince's work in this global fight against what is both a national security threat and a devastating environmental problem" the administration wrote in a press release.

The trip will mark Prince William's first visit to the nation's capitol.

@highlight

The President will meet with Prince William on Monday

@highlight

The Duke of Cambridge will also meet with Joe and Jill Biden

@highlight

William and his wife, Kate Middleton, will be in the U.S. for three days

摘要

项目效果

输入关键词：

google

开始检索　关键词分析

关键词	出现总次数	出现的文档数	idf值
google	121.0	38	3.963316299815697

下列是以关键词 **google**进行检索返回的结果：

本次查询共用时 1.3148 秒，为您找到 38 篇相关新闻， 104 相关句子

新闻ID：1815 相似度：0.4817 相关句子：12

- 1 (cnn) -- **google** thursday released latest candy-themed mobile operating system : android 4.4 , deliciously known kit kat .
- 2 system launch immediately **google** 's new nexus 5 phone roll nexus device , samsung galaxy s4 htc one , next week .
- 3 nexus device **google** 's flagship line phone tablet , running newest purest form android available without contract . nexus 5 , manufactured lg , built work internationally variety band whatever local sim card pick , though verizon 's band still supported .

查询过程

- 输入关键词
- 根据查询与新闻的相似度返回新闻。

展示页面包括

- 关键词的idf值
- 查询用时
- 返回的新闻与查询的相似度
- 新闻中与之匹配的句子

项目效果

新闻ID：1949 相似度：0.0637 相关句子：2

- 25 last month , friend two british skier went missing verbier swiss alp used twitter find friend ' cell phone number . helped mountain rescue team locate one missing skier using gps **google** map cell phone .
- 26 **google** 's latitude service , launched february , constantly update user ' location , enabling contact see whereabouts map well activity via status update .

其他信息：该新闻(查看原文)共有34个句子，共计678个词。

新闻ID：1758 相似度：0.0530 相关句子：2

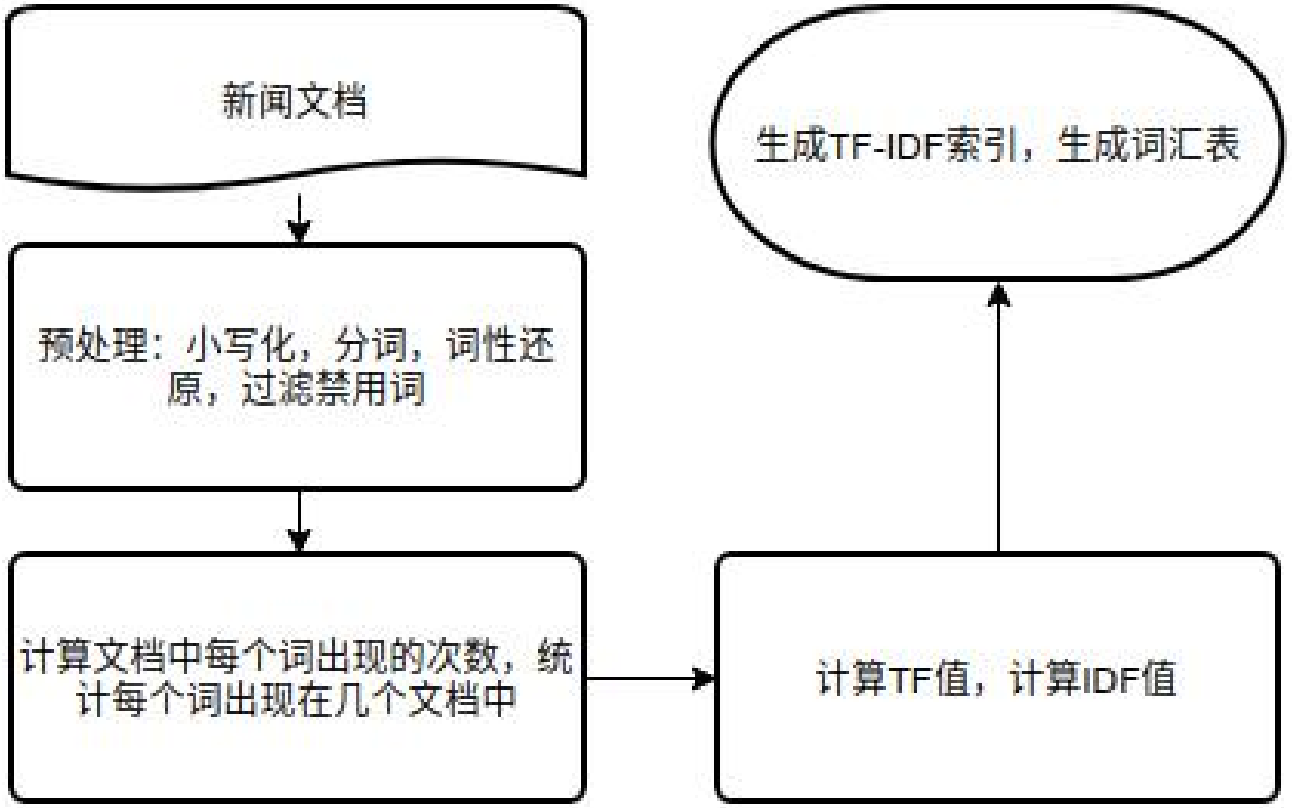
- 32 **google** , part , offer taiwan `` palestinian territory , '' well western sahara , breakaway region morocco recognized internationally .
- 34 neither yahoo **google** responded repeated cnn request comment .

其他信息：该新闻(查看原文)共有40个句子，共计635个词。

实现步骤

- 处理文档，计算tf-idf索引，将每个文档映射为一个单位向量
- 处理查询，将查询映射为一个向量
- 通过内积计算查询向量与文档向量的相似度
- 返回相似度大于0的文档

实现步骤：索引的构建



- 词性还原：fish，fishes，fishing，fisher，fished => fish
- 禁用词过滤：几乎在所有文档中出现的词，没有检索价值，比如“the”，“a”，“an”
- TF-IDF索引：列是新闻，行是每一个单词，列向量代表某一个文档。
- 对列向量进行归一化，这样与同样归一化的查询向量做内积的时候，结果就是两个向量夹角的余弦值（余弦相似度）

实现步骤：将查询映射为向量

- 加载第一步保存的词汇表，初始化一个词汇表长度的零向量
- 根据第一步中的预处理步骤，对查询进行预处理
- 对于查询中的每个关键词，将向量对应的位置设置为1

所用工具

- 编程语言：python
- 文档预处理：nltk
- 索引的构建、操作：pandas，numpy
- 网页前端页面：Boostrap
- 网页后端：bottle

项目目录结构：

观察1：验证ZIPF定律

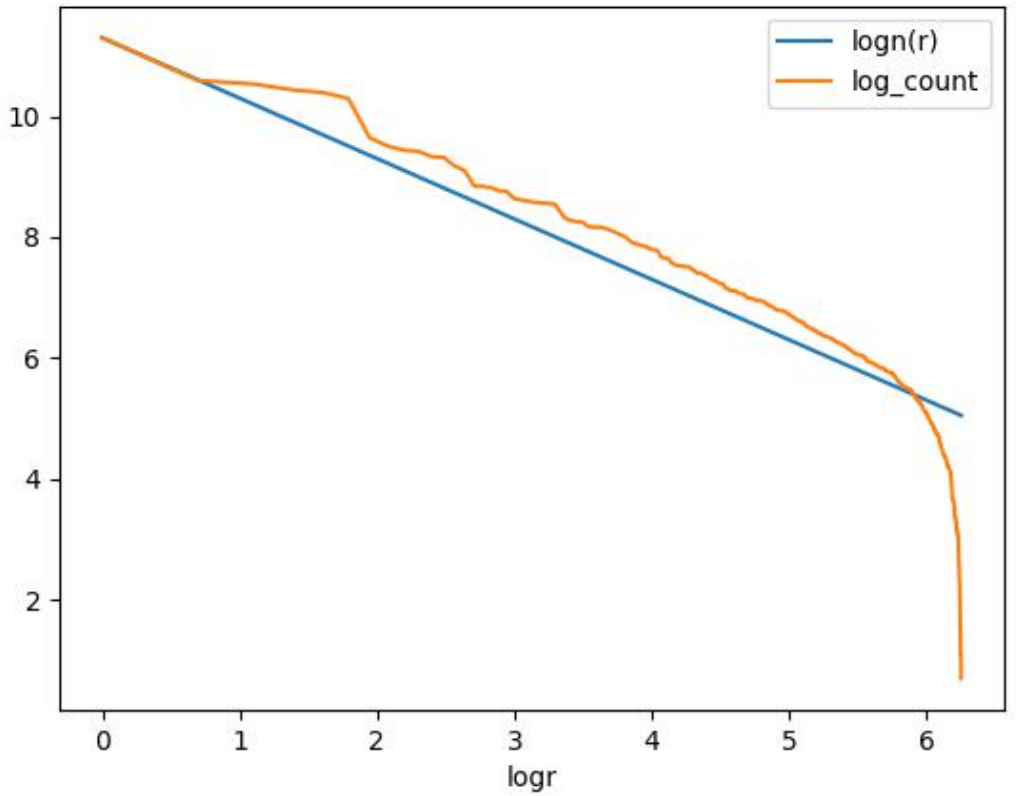
Zipf定律：在自然语言的正文中，词语的频率可以由数据概率分布来估计

$$logn(r) = logC - logr$$

r：单词出现的频率排名

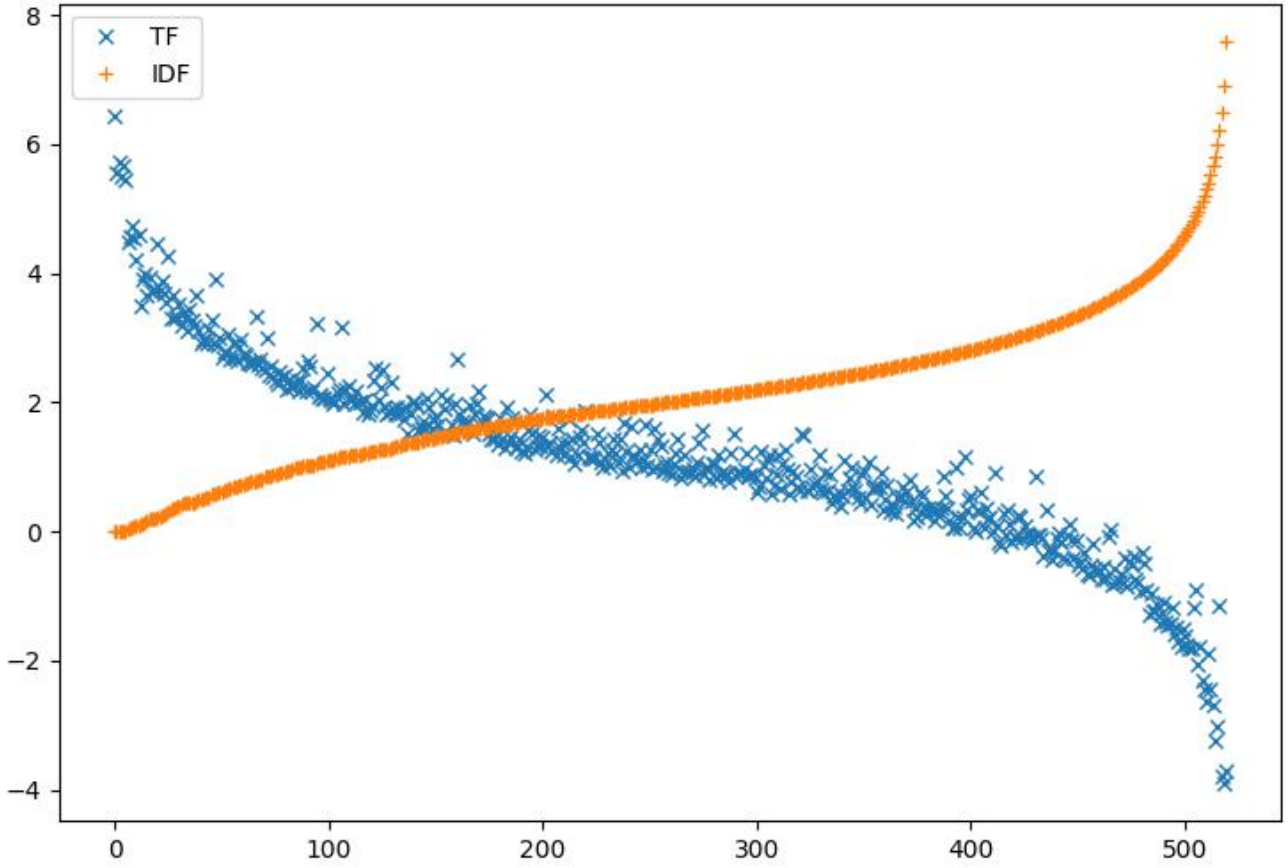
$n(r)$：该排名对应的单词频率

C：常数项，$r = 1$时可得$C = n(1)$



基本近似

观察2：TF-IDF的性质



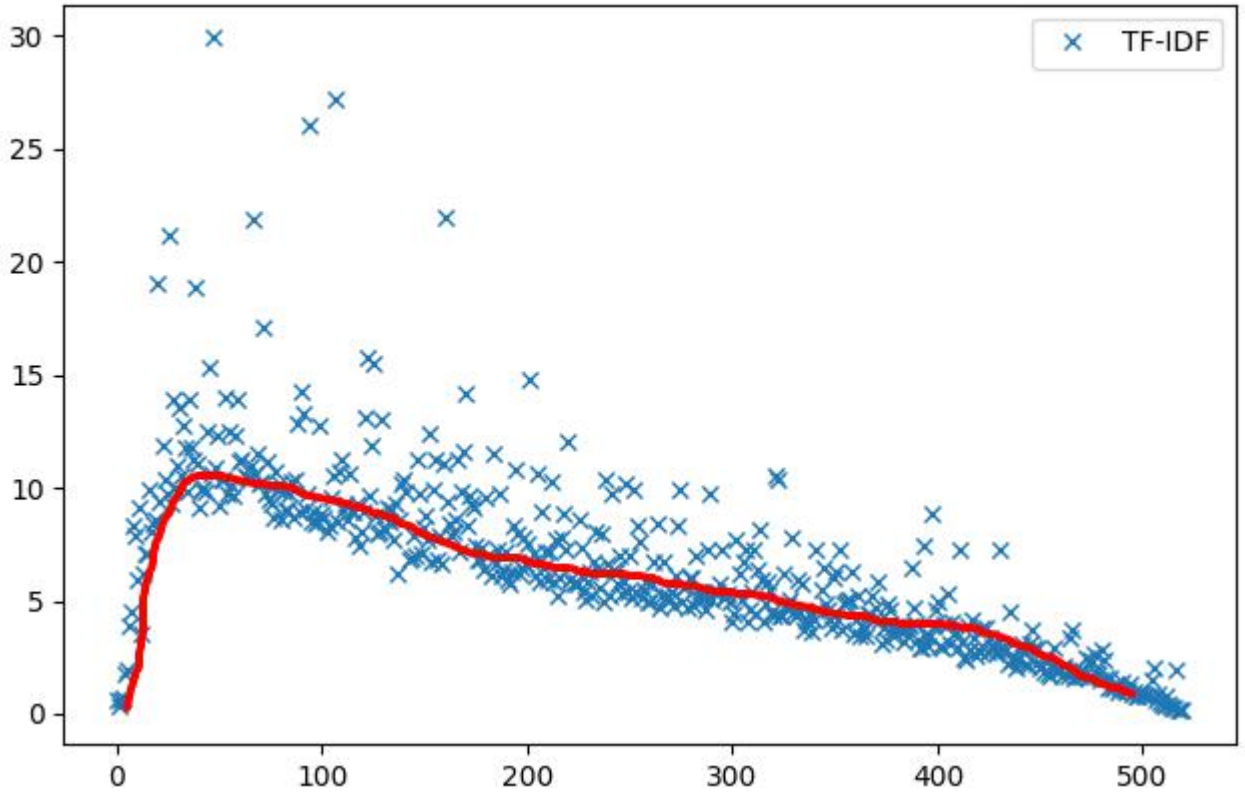
$TF_i = log \sum_{j=1}^{N} tf_{i,j}$

横轴是词的编号i，

纵轴是词对应的 TF_i 以及 IDF_i 的值。

- TF 和 IDF 权重会相互平衡
- 高TF 对应低 IDF
- 低 IDF 对应高 TF

观察2：TF-IDF的性质



结果就是含有中等 IDF 的单词

往往对应 最大的 $TF-IDF$ 权重，

这说明，像具有低 IDF 权重的高频词，

或者是像错误拼写那样的高 IDF 权重的词，

对于排序都没有很大价值（就是对相似度的贡献很小）。

参考自 **现代信息检索**

优点以及不足之处

- 优点：基于TF-IDF的向量模型简单，容易实现，检索质量较好。

- 不足：
  - 索引项被假定是相互独立的，丢失了索引的位置信息。
  - 难以进行查询拓展，实现更复杂的查询。
  - 如果有新的文档，更新索引需要很大的计算量。

值得改进的地方

- 实现复杂查询（比如“word1 word2” => 限定顺序 ，可以在向量模型返回的结果中继续进行过滤。）
- 索引的计算，相似度的计算，可以并行化（cupy可以借助GPU进行加速）

- 尝试其他模型：
  - BM25模型
  - 语言模型 ==========>>
  - 潜在语义索引模型

为每个文档构建一个概率语言模型 M_d ，

计算该语言模型生成对应查询的概率 $P(q|M_d)$，

这个概率可以看成文档与查询的相关度。