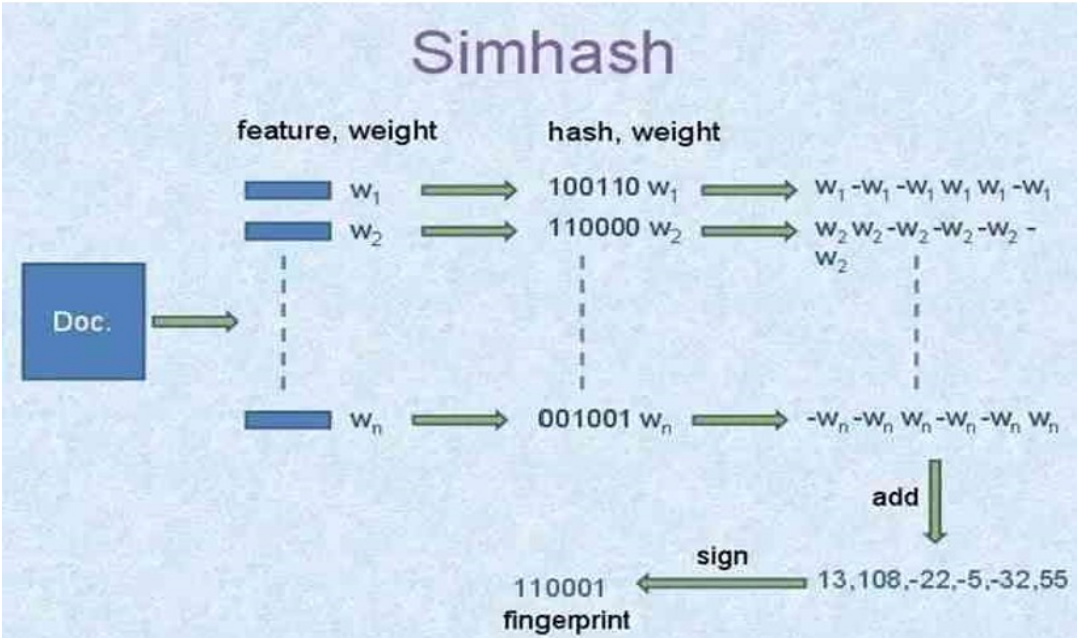
【问题描述】

谷歌、百度等大型搜索引擎需要定期抓取全球网站的网页数据，并建立索引用于支持关键词的快速查询。通常搜索引擎需要判断抓取的网页是否在已经索引的数据集中，来决定是否更新索引，从而提高搜索引擎的更新效率。这需要高效文本去重算法进行支持。

Simhash 是 Google 用来处理海量文本去重的算法（参考论文《Detecting Near-duplicates for web crawling》），借助于文本向量的构建以及局部敏感哈希技术，可以将每个网页文本生成一串二进制编码（文本指纹 fingerprint），再利用高效的索引方法，加速大规模文本数据集中相似文本的检测。网页指纹生成原理如下图所示：



1．首先统计网页（Doc）特征向量 feature 中每个特征的权重（$w_1, w_2, \cdots w_n$）（如何确定特征及其权重详见下面的具体实现方法）；

2．依据每个特征对应的哈希值（hashvalue，一个由 01 组成的长度为 M 的串，其中 1 表示 1，0 表示-1，M 为指纹长度）得到一组符号化权重值。如图所示，“100110”为长度为 6 的哈希串，“100110 w_1”将得到一组（6 个）符号化 w_1 值“w_1 -w_1 -w_1 w_1 w_1 -w_1”。

3．特征向量所有特征对应位置符号化权重值相加，将得到一组整数，如图所示“13, 108, -22, -5, -32, 55”。

4．获取上步得到的一组整数的符号值（1 表示正数，0 表示负数和零），将其组成符号值串，即为该网页的指纹（fingerprint）。如图所示，“13, 108, -22, -5, -32, 55”其符号值串为“110001”，也就是网页（Doc）的指纹。

基于 Simhash 原理实现一个相似网页（文本）检测工具，方法如下：

1．获取网页特征向量。对所有网页（文本）的非停用词（stopword）英文单词进行词频（出现次数）统计，并将单词词频由高到低进行排序（频度相同时，按字典序），取前 N 个单词构成网页特征向量 feature=（$word_1, word_2, \cdots, word_N$）。N 为特征向量维度，或长度。

注意：英文单词为仅由字母构成的字符串，不区分大小写。统计时要将大写字母转换为小写字母。在自然语言处理中，停用词（stop-word）指的是文本分析时不会提供额外语义信息的词的列表，如英文单词 a，an，he，you 等就是停用词。

2．统计每个网页（文本）的特征向量中每个特征（单词）的频度（权重），得到特征向量对

应的权重向量 weight=（$w_1, w_2, \cdots, w_N$）。
3．每个特征 $word_i$ 均有一个对应哈希值串 $hash_i$，每个网页的特征向量对应的权重向量中权重 w_i（i=1,2,⋯,N）按对应哈希值串 $hash_i$ 进行符号取值，得到一组由 w_i 和-w_i 组成的符号化权重值向量 $SignWeight_i$。（权重值为 0 时符号化后值仍为 0）。
4．计算网页指纹 fingeprint。对每个网页，特征向量的所有特征对应的符号化权重向量对应位置值累加，并对累加结果，大于 0 置 1，小于等于 0 置 0，得到网页（文本）指纹（fingerprint）（一个由 01 组成的长度为 M 的编码串串）

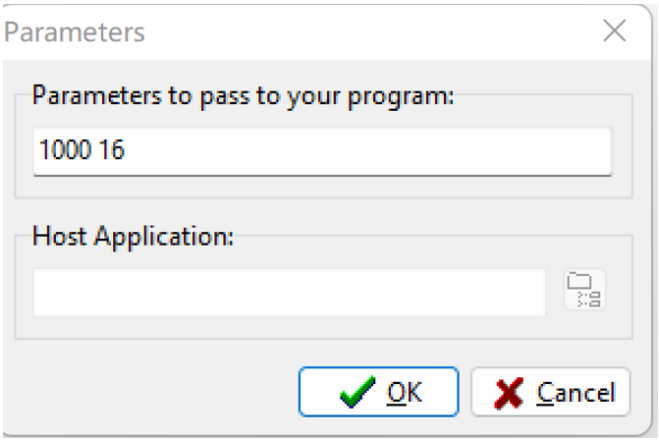
$$fingeprint = (Sign(\sum_{i=1}^{N} SignWeight_{i1}), Sign(\sum_{i=1}^{N} SignWeight_{i2}), \ldots, Sign(\sum_{i=1}^{N} SignWeight_{iM}))$$

其中：
Sign(X) = { 1 | 0，当 X>0 时为 1，X<=0 时为 0}
5．按上面方法计算新抓取的网页的指纹。
6．基于文本指纹相似度可以对新抓取的网页与已有的网页数据集进行相似比较。指纹相似度可通过汉明距离（Hamming distance）进行计算：两个文本指纹的汉明距离是指其二进制编码串中 01 取值不同的数量。举例如下：文本指纹 10101 和 00110 从第 1 位开始依次有第 1、第 4、第 5 位不同，海明距离为 3。
基于汉明距离可以筛选出大概率相同的网页文本，一般超过某个阈值（在此，阈值设置为 3）则判定为不相似，小于等于阈值判定为相似。
7．按输出形式要求，依次将新抓取网页与已有相似网页（指纹汉明距离阈值为 3 以内的）按汉明距离由小到大网页输出到屏幕和指定文件中。

【输入形式】
从命令行输入特征向量长度 N 以及指纹长度 M。
具体形式如下：
simtool N M
其中 simtool 为网页相似检测程序。其根据当前目录下的停用词文件“stopwords.txt”、哈希值文件“hashvalue.txt”、已有网页数据文件“article.txt”、待查重（即新提取）的网页数据文件“sample.txt”，按上面要求依次对 sample.txt 中每个网页文档在 article.txt 文件中查找相似网页，并按输出要求输出检测结果。
注意：
1．在课程网站“课件下载”区提供了 project2023.zip，其中包括英文停用词表“stopwords.txt”文件(文件中只包含单词，不含其解释，且已按字典序排序）和哈希值文件“hashvalue.txt”。该 hashvalue.txt 文件中包含了 10000（行）x128（列）由 01 组成的数据，每一行为一个哈希值串，若程序命令行输入的特征向量 N 为 1000 和指纹长度 M 为 16 时，则取该文件前 1000 行，每行取前 16 列作为实际哈希值表在程序中使用。**哈希值文件的规模决定了本题的特征向量最大长度不超过 10000，指纹最大长度不超过 128。**
2．由于 Windows 系统下文本文件中的'\n'回车符在（评测环境）Linux 系统下会变为'\r'和'\n'2 个字符，建议用 fscanf(fp,"%s",⋯)来处理停用词文件中英文单词。
3．为了简化相似检测程序的实现，已从互联网上爬取（Web Crawling）相关网页(文档)的工作已经完成，并将爬取的网页文档数据已存入一个文本文件（aritcle.txt）中，其中每个网页第一行为网页标识号（如 XX-XXXX，可按字符串来输入），然后为网页内容，网页文档间以换页符\f 分隔。在课程网站下载区提供了一个用于测试的 article.txt 文件。sample.txt 文件中每个网页的标识号为“Sample-XXX”，其它格式同 article.txt 文件。

【输出形式】
按下面形式依次输出 sample.txt 文件中每个网页在 article.txt 中找到的相似网页信息到文本文件 result.txt 中：
X1
0:ID01 ID02 …
1: ID11 ID12 …
2: ID21 ID22 …
3: ID31 ID32 …
X2
…
其中 X1,X2…为 sample.txt 中网页标识号，次序同原文件中序；“0:ID01 ID02 …”表示在 article.txt 中与相应网页指纹汉明距离为 0 的所有网页标识号，按 article.txt 文件中出现序排列，中间以一个**空格分隔**。**若不存在相关网页，则无汉明距离为 0 的信息输出，即无“0:…”一行信息输出，其它部分含义相同**。汉明距离相同的**最后一个网页标识后也有一个空格分隔符**。行末换行时输出字符“\n”即可。
同时，将 sample.txt 文件中第一个网页相似检测结果信息输出到屏幕上，即输出下面信息到屏幕上：
X1
0:ID01 ID02 …
1: ID11 ID12 …
2: ID21 ID22 …
3: ID31 ID32 …

【样例输入】
假设 simtool.exe 为网页相似检测程序，以下面方式运行该程序：
simtool 1000 16
（运行程序前，从课程网站下载区下载 project2023.zip 文件，其中包括：article.txt, sample.txt, hashvalue.txt, stopwords.txt, results(example).txt）
说明：若本地编程环境为 dev-C++，可点击菜单 Execute\Parameters…，在下面对话框中输入相应命令行参数。



【样例输出】

假设 simtool.exe 为网页相似检测程序，以下面方式（特征向量长度为 1000，指纹长度为 16）运行该程序：

simtool 1000 16

程序运行后，屏幕上输出的结果为：

```
Sample-1
0:1-1
1:1-111 1-327 1-901 1-917
2:1-79 1-148 1-200 1-235 1-319 1-353 1-380 1-381 1-391 1-508 1-511 1-531 1-571 1-577 1-614 1-616 1-838 1-842 1-872 1-959
3:1-10 1-43 1-51 1-62 1-71 1-77 1-91 1-93 1-96 1-103 1-114 1-115 1-116 1-132 1-155 1-156 1-158 1-165 1-170 1-193 1-203 1-206 1-217
 1-225 1-226 1-233 1-241 1-246 1-253 1-296 1-315 1-328 1-345 1-368 1-400 1-407 1-446 1-453 1-458 1-480 1-494 1-496 1-502 1-509 1-5
10 1-512 1-522 1-543 1-552 1-553 1-559 1-572 1-595 1-597 1-666 1-674 1-675 1-679 1-695 1-696 1-712 1-727 1-729 1-730 1-743 1-796 1
-797 1-801 1-813 1-830 1-835 1-846 1-849 1-851 1-868 1-873 1-883 1-892 1-902 1-905 1-923 1-944 1-945 1-955 1-963 1-964 1-972 1-975
 1-976 1-990

--------------------------------
Process exited after 0.4416 seconds with return value 0
请按任意键继续. . .
```

所生成的结果文件“result.txt”内容应与下载区文件“result(example).txt”完全相同。

若以下面方式（特征向量长度为 1000，指纹长度为 32）运行该程序：

simtool 1000 32

程序运行后，屏幕上输出的结果为：

```
Sample-1
0:1-1
2:1-901


--------------------------------
Process exited after 0.2058 seconds with return value 0
请按任意键继续. . .
```

【样例说明】

以上面屏幕输出为例，其中第一行 Sample-1 为 sample.txt 中第一个网页标识号；第二行开始输出汉明距离（值为 0，1，2，3）及在文件 article.txt 中与给定网页汉明距离为相应值的网页编号，网页编号间以一个空格分隔。若 article.txt 中不存在与给定网页汉明距离为某个值的网页，则该行不输出，如上面第 2 个屏幕输出，由于不存在汉明距离为 1 和 3 的网页，则该行不输出。输出文件 result.txt 中信息含意与此类同。

【评分标准】

本题为综合功能测试题，其评分标准为通过测试数据即可得分。程序运行无结果或结果错误将不得分。