

**信息与知识获取**



**信息检索系统**

**姓 名 黄链泽 鄭毓恒 刘言**

**学 院 计算机学院**

**专 业 计算机科学与技术**

**班 级 2020211302**

**学 号 2020211246**

**2020211262**

**2020211268**

**任课教师 乔秀全**

**2023年 5 月**

**作业要求**

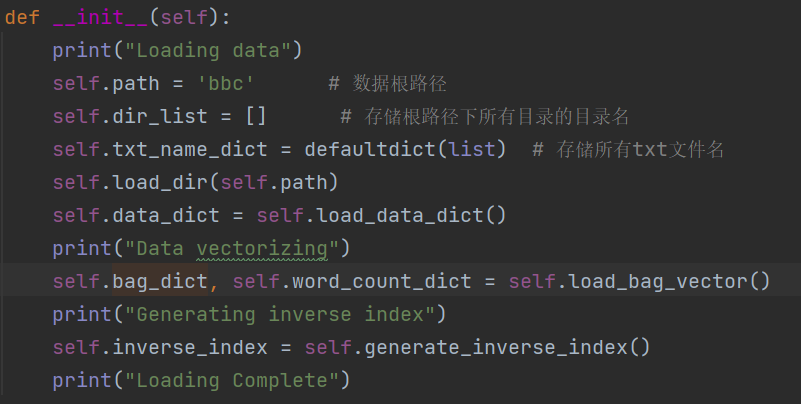
自己动手设计实现一个信息检索系统，中、英文皆可，数据源可以自选，数据通过开源的网络爬虫获取，规模不低于100篇文档，进行本地存储。中文可以分词（可用开源代码），也可以不分词，直接使用字作为基本单元。英文可以直接通过空格分隔。构建基本的倒排索引文件。实现基本的向量空间检索模型的匹配算法。用户查询输入可以是自然语言字串，查询结果输出按相关度从大到小排序，列出相关度、题目、主要匹配内容、URL、日期等信息。最好能对检索结果的准确率进行人工评价。界面不做强制要求，可以是命令行，也可以是可操作的界面。提交作业报告和源代码。

**具体实现**

本次作业使用网上现有的数据源，从[ML Resources - BBC Datasets (ucd.ie)](http://mlg.ucd.ie/datasets/bbc.html)下载。数据源来自BBC网站，包含了由2004至2005年有关于商业、娱乐、政治、运动和科技五个主题的新闻，总共有2225个TXT文档。

本次作业采用了scikit-learn库中的Bag of Word模型构建文档向量，将单词之间的顺序关系全部丢弃，只关注文档中单词出现的次数。Bag of Word模型为每个单词分配一个唯一编号，这样一个文档就能够被编码成与已知词汇数量相同长度的向量。向量中每一个位置上的值就是其编号对应单词在文档中出现的次数。

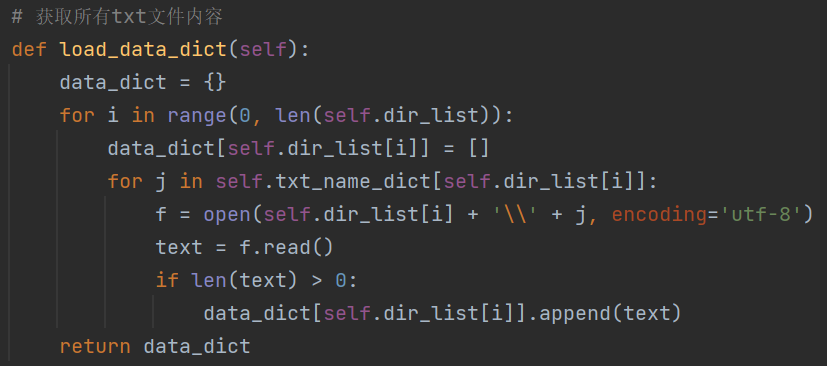
代码为向量空间检索模型定义了一个InfoRetrie类，类的初始化函数如下：



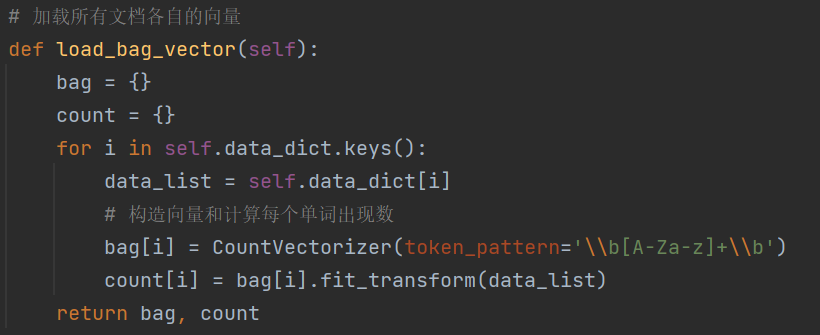
函数首先调用成员函数load\_dir()，加载数据文件夹根路径bbc中所有子目录名和文件名，分别存储到类成员dir\_list和txt\_name\_dict当中。dir\_list是字符串类型的list，存储所有子目录路径。txt\_name\_dict索引为目录路径，值类型为list，存储该路径下所有txt文件名。函数定义如下：



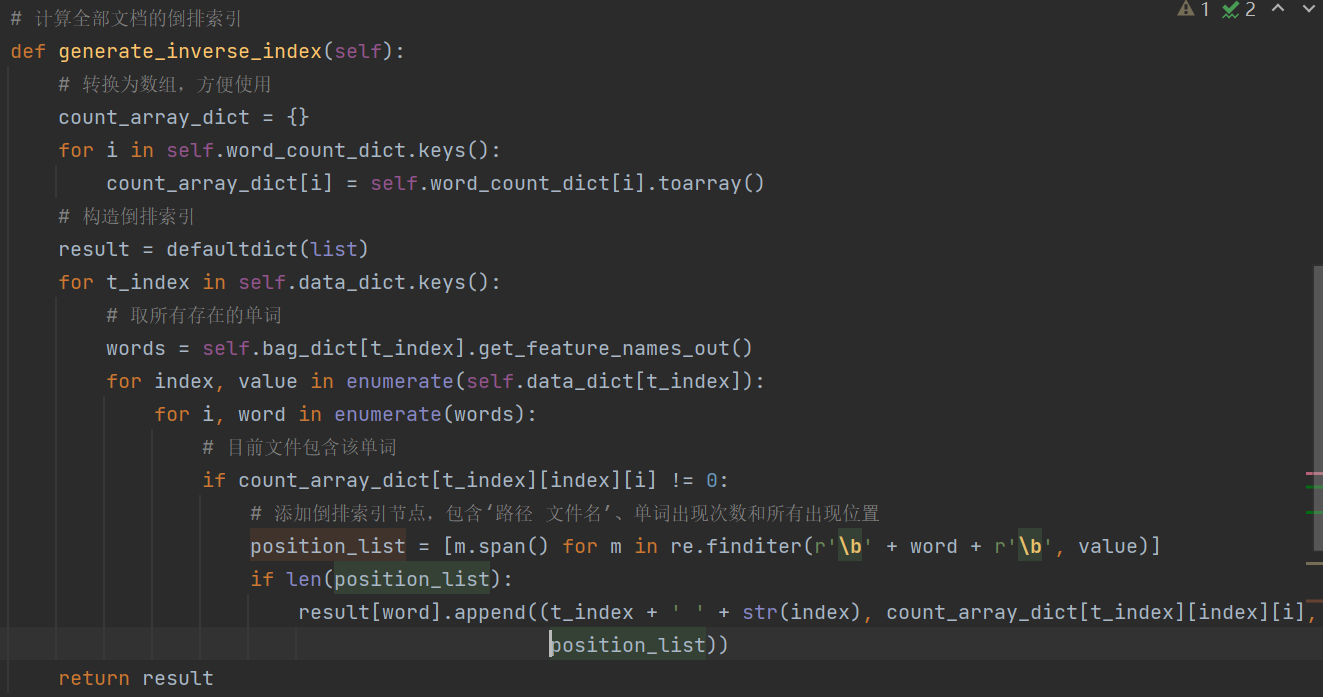
然后，调用load\_data\_dict()函数，通过dir\_list和txt\_name\_dict的存储内容，读取所有txt文件，存储到data\_dict当中。data\_dict的索引为目录路径，值类型为list，存储该目录下所有txt文件的内容。函数定义如下：



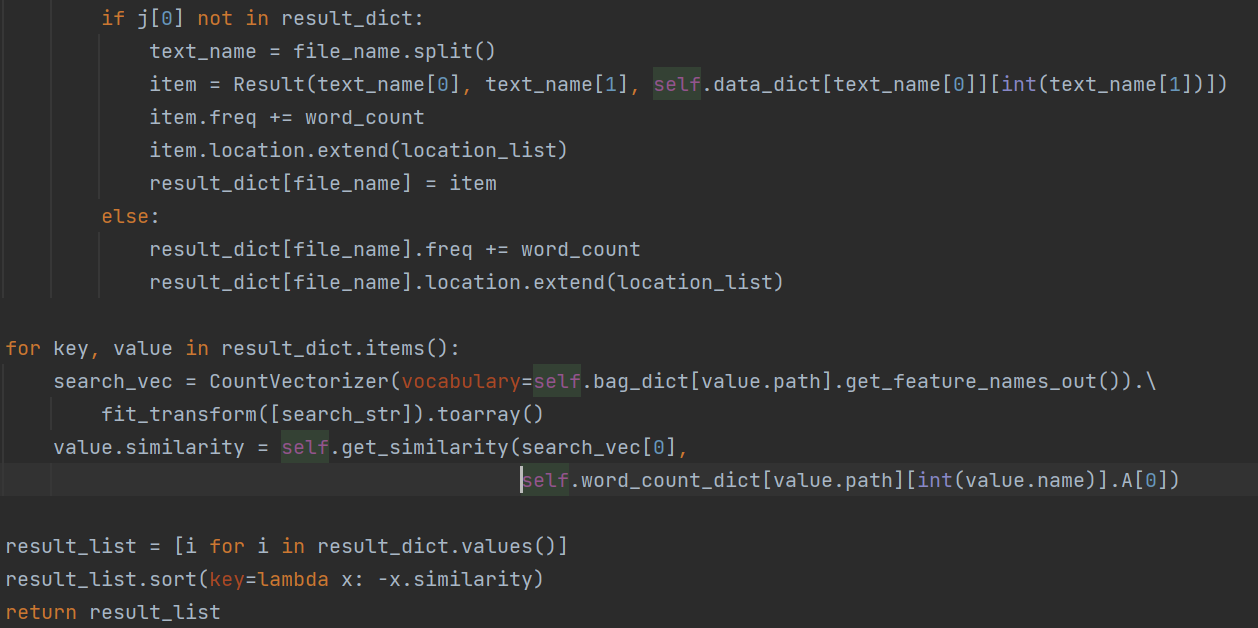
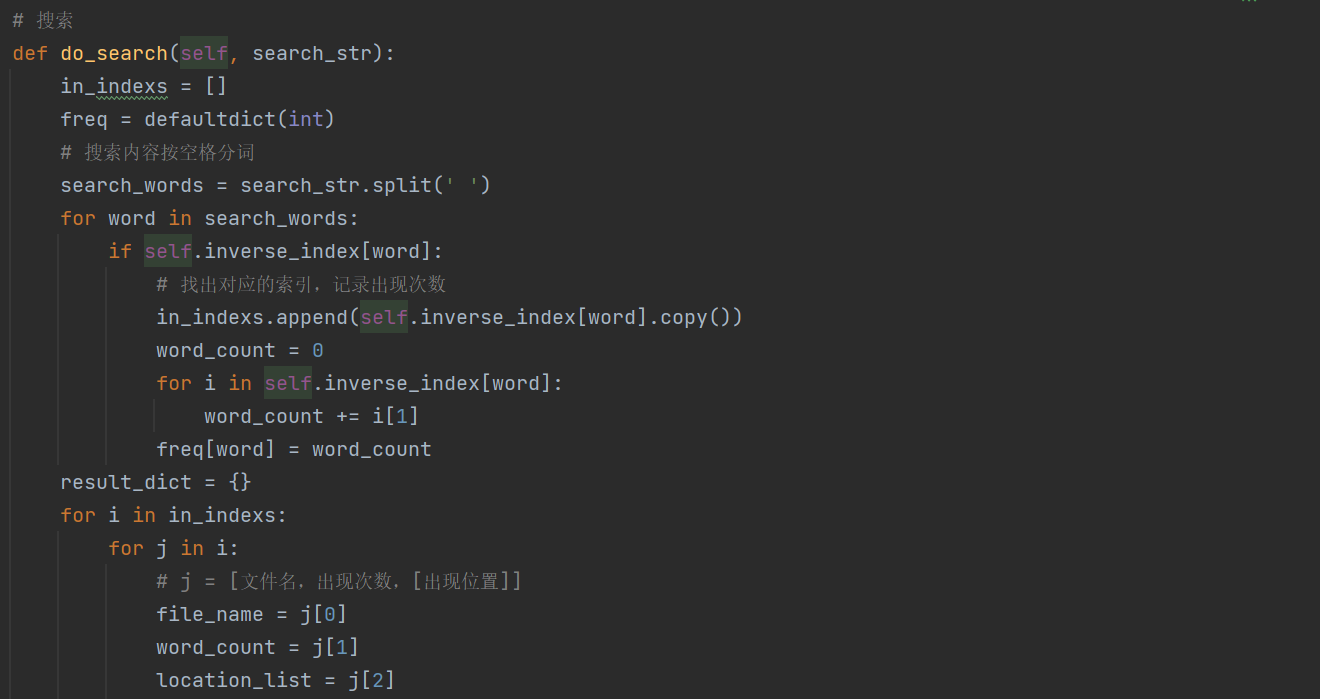
在函数load\_bag\_vector()，利用data\_dict的内容，分别为每个子目录，根据所在该目录的文件内容，构造bag of word模型。本次作业使用CountVectorizer将文本转换为词统计向量，向量中每个位置对应词汇表该位置上的词在当前文件中出现的次数。结果词汇表存储在bag\_dict，词统计向量存储在word\_count\_dict，它们的索引均为目录路径。函数定义如下



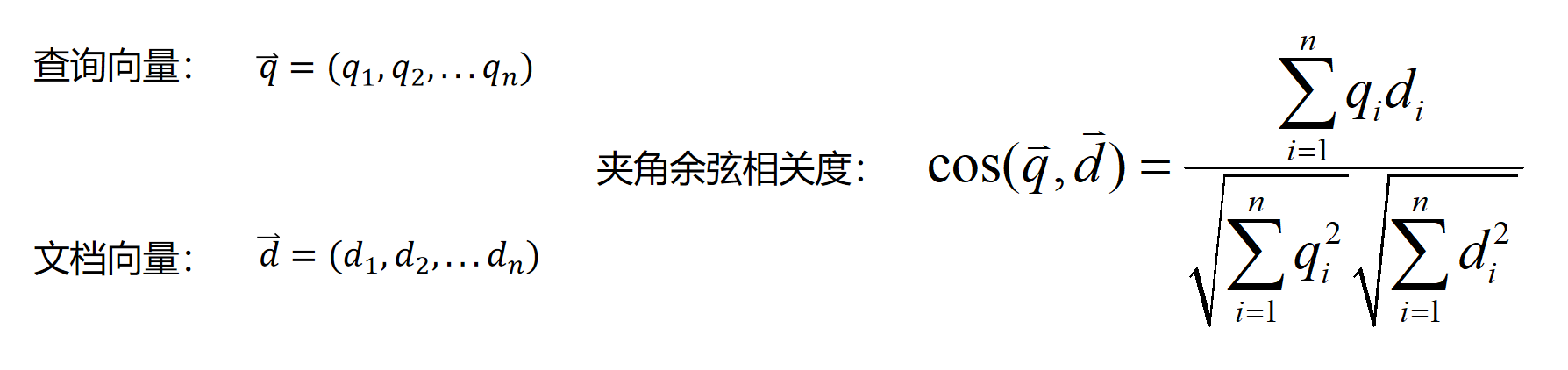
在函数generate\_inverse\_index()，构造倒排索引，存储在类成员inverse\_index。该成员的类型是一个dict，索引是单词，值类型是一个存储三元tuple的list，tuple包含‘路径 文件名’、单词出现次数和所有出现位置。函数定义如下：



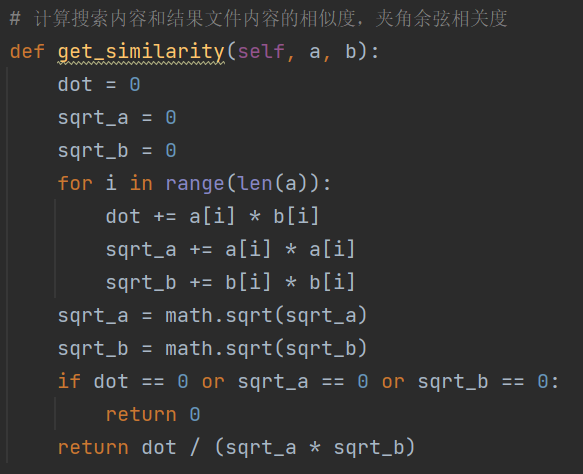
至此，构造完向量空间以及倒排索引，可以通过调用do\_search函数，进行检索。函数首先对检索内容按空格进行分词，然后对每个词，在倒排索引中找出对应的所有节点，得到包含该词的所有文件以及词在文件的出现位置，统计总共出现次数。接着，把搜索结果存储到Result类中，按结果文件和检索内容的相关度进行排序，返回以list存储的最终结果。函数定义如下：



InfoRetrie的成员函数get\_similarity负责计算两个搜索内容和结果文件内容的相关度，使用的是夹角余弦相关度。这个算法利用向量之间的夹角来衡量两个文档之间的相关度。夹角越小，说明两个向量所代表的文档越相关；夹角越大，则说明两个向量所代表的文档越不相关。



函数定义如下：

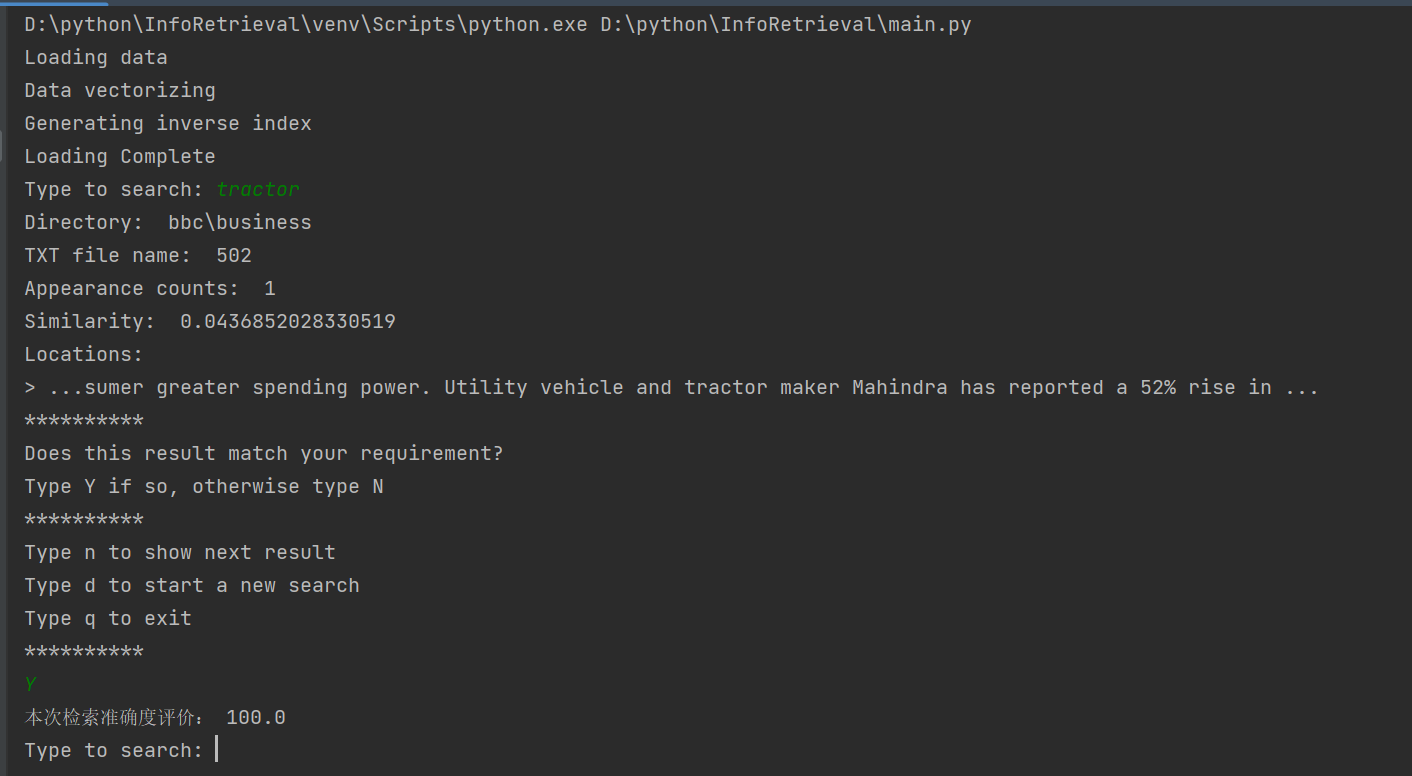


存储结果的自定义Result类包含搜索结果文件路径、文件名、文件内容、检索内容出现次数、所有出现位置和相关度，并且包含一个输出函数。函数定义如下：

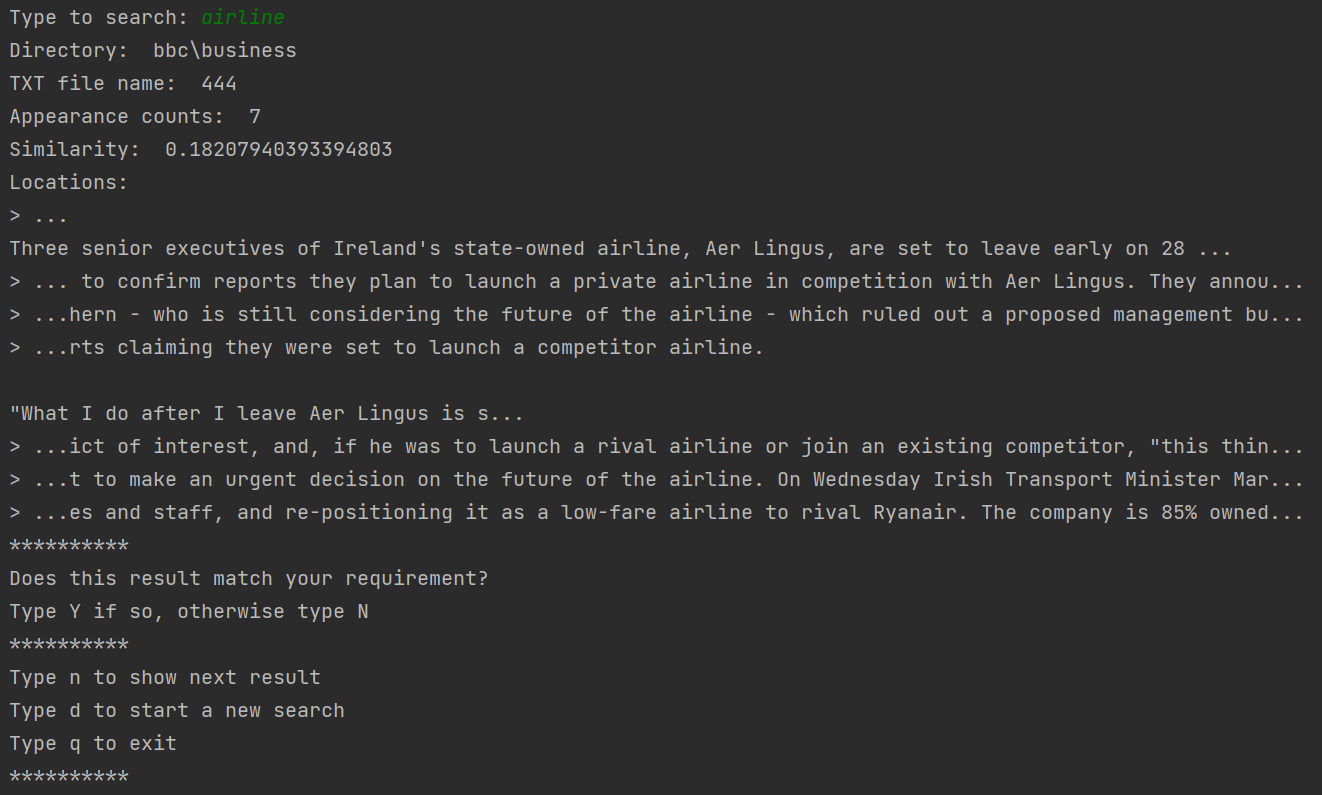


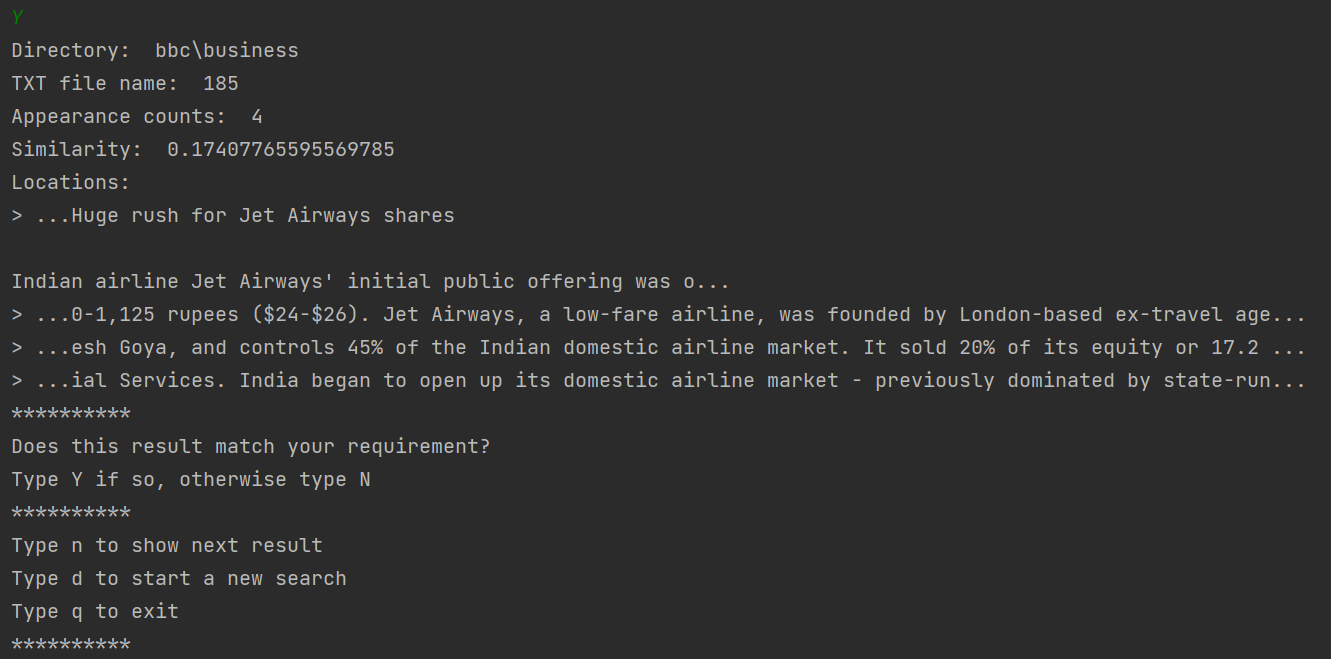
**运行结果**

尝试搜索单词tractor，输出了以下结果，包含了文件路径、文件名、在文件的出现次数、相关度和出现位置。输出一个结果后，可以输出Y或N表示对这个结果满意与否，不输入评价则默认为Y，评价后自动输出下一个结果，也可以输入d进行另一次搜索或者输入q退出程序。结束本次搜索后，会输出本次的用户好评百分比（满意的结果/输出结果总数）。

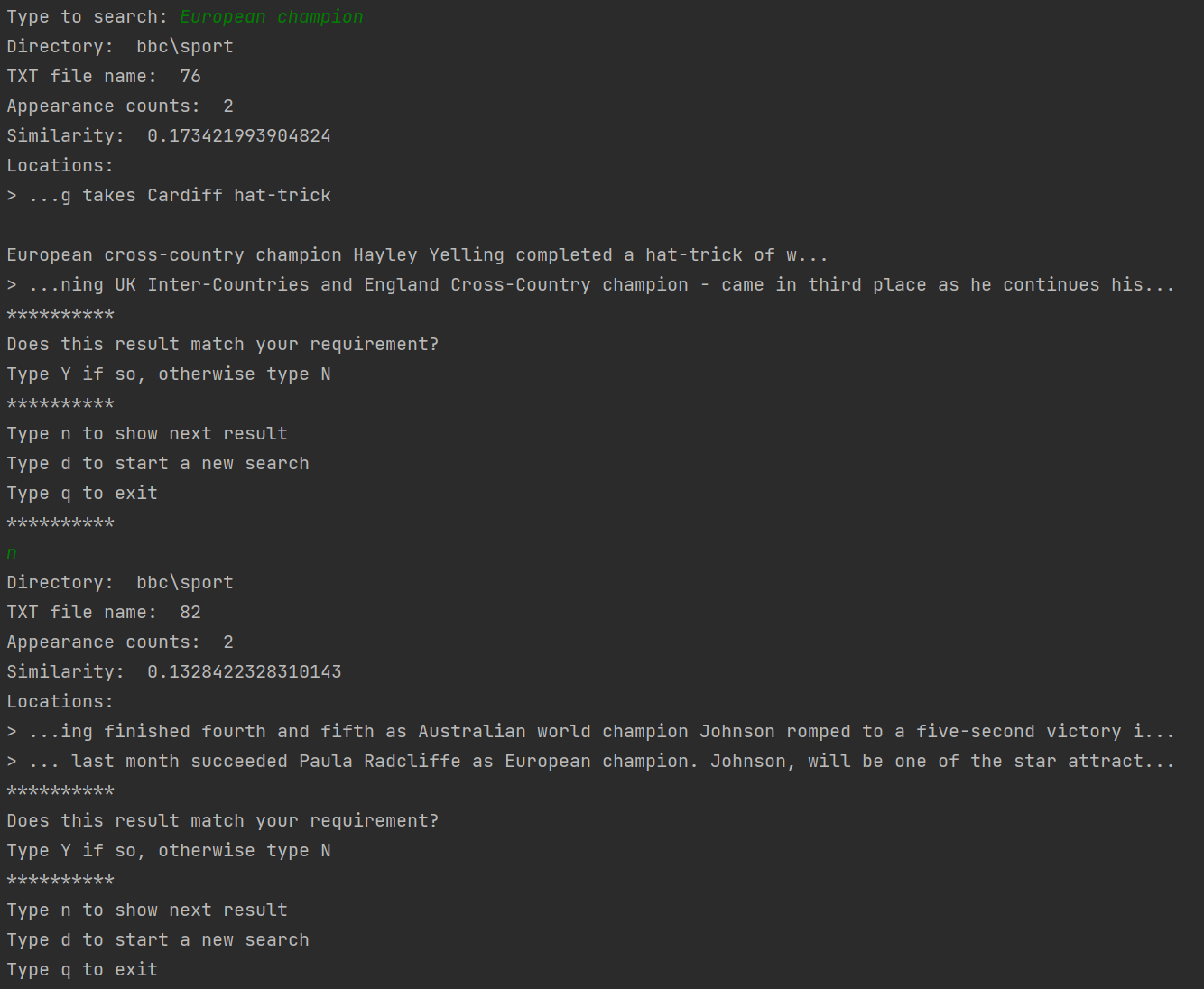


尝试搜索airline，可以按相关度从大到小输出结果。





尝试搜索European Champion，同样可以按相关度从大到小输出结果。



**遇到的问题**

经过测试，每次运行程序都需要很长时间加载倒排索引。尝试过在第一次运行程序后，将向量空间模型和倒排索引存储到文件中，但由于数据结构比较复杂，最终没有实现。随着文件增多，加载的时间会更长，不方便使用。

程序支持多单词搜索，但很多时候，即使相关度高，查找结果也只包含部分单词。其中一个原因是比起整个要搜索的字符串，部分单词出现次数更高，例如单词the和to，导致程序输出一些并不相关的结果。这是一个可以改进的部分。