**top-k空间文本簇检索系统架构设计**

空间文本簇检索过程包括文本预处理、空间索引构建、文本相似性度量、聚类算法选择、结果评价等过程。文本预处理阶段分为文本分词、构建倒排索引表；空间索引结构采用 IR-Tree 结构；文本相似度为余弦相似度；聚类算法选择基于密度的聚类方法中最具代表性的 DBSCAN 算法。总体设计流程如图[2.1](#_bookmark10)所示：

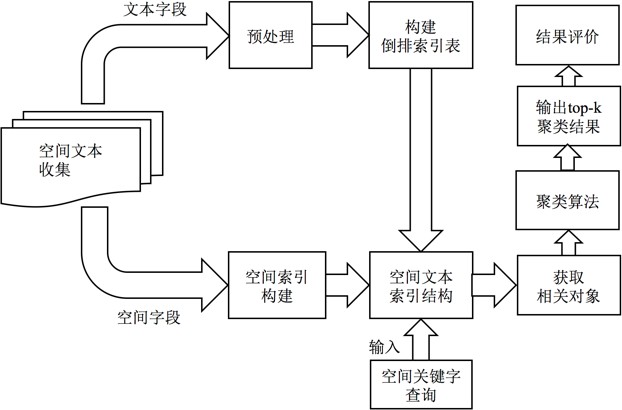


图 2.1: top-k 空间文本簇检索设计流程

本论文的空间文本簇检索系统架构按模块在大方向上分为数据处理模块和检索模块，数据处理模块又细分为分词模块、文本向量化模块、构建空间索引模块，分词模块调用了两个外部接口。分词模块和文本向量化模块将数据对象的文本信息转化为结构化信息，和由数据对象的空间信息构成的空间索引结构 R-Tree 一起构成空间文本索引结构 IR-Tree。检索模块依据空间关键字在 IR-Tree 上查询邻域内相关对象，查找满足条件的簇。总体的系统架构图如图[2.2](#_bookmark17)所示：

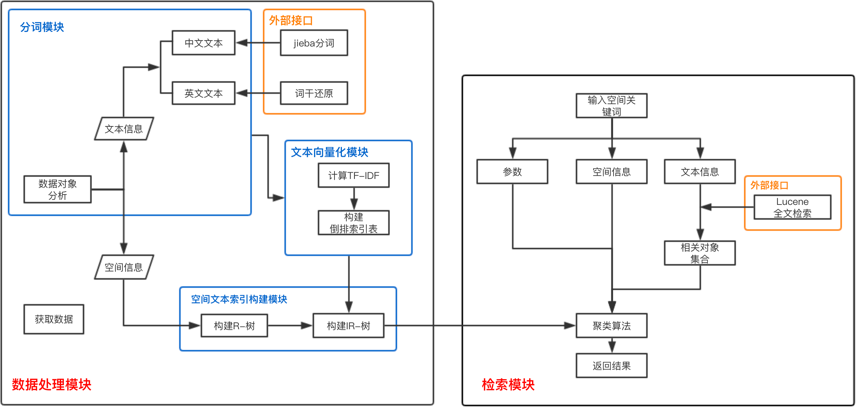


图 2.2: top-k 空间文本簇系统框架图

**1. 数据处理模块**

网络文本通常呈现出非结构化，不能被计算机直接处理。所以聚类首先需要将收集到的网络文本转换为结构化的数字信息，即空间文本表示模型。本论文采用向量空间模型（vector space model， 简称 VSM），向量空间模型是信息检索的基础，包括文档分类和聚类等，是目前数据挖掘中最常用的表示模型。另外常用的表示模型还有布尔模型和概率模型等。

数据处理模块通过文本分析将数据对象的文本描述信息传递给分词模块处理，分词模块根据中英文作出相应的分词处理；文本向量化模块则计算出分词后每个词项在文本中的权重，并构建倒索引表。完成对单个文本的向量化处理后，构建向量空间模型，称为反向文件。最后将空间字段和反向文件传递给空间文本构建模块，建立IR-Tree。下面对每个小模块作进一步说明：

**1.分词模块:** 文本中，英文的表示以单词作为基本单位，单词之间以空格作为分隔符。根据分隔符划分文本并去除停用词后得到的词项结构已经可以达到了不错的分词效果。

另一方面，对于中文分词，情况要复杂的多。中文没有直接的分隔符，灵活性和复杂性要比英文高得多。本研究将使用 github 上一种基于统计分词方法的开源中文分词项目 jieba 分词，对中文文本进行分词处理。

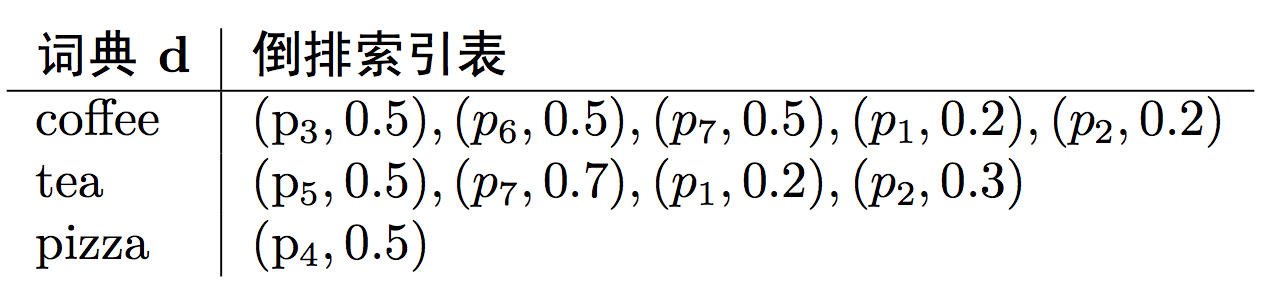
分词模块将数据集的每一个文档按照文本字段和空间字段分别处理，文本字段即对象的描述内容，空间字段即对象的位置信息。分词模块会对分开处理中文文本和英文文本，对于中文文本，模块调用 jieba 分词接口进行分词处理，英文文本根据分隔符对单词进行预处理，再调用词干还原接口对单词作进一步处理。分词结果以词项列表的形式存储。通过接口处理完文本后，每个文本的词项都存储在内存中，并以传递给文本向量化模块。分词部分代码如下：



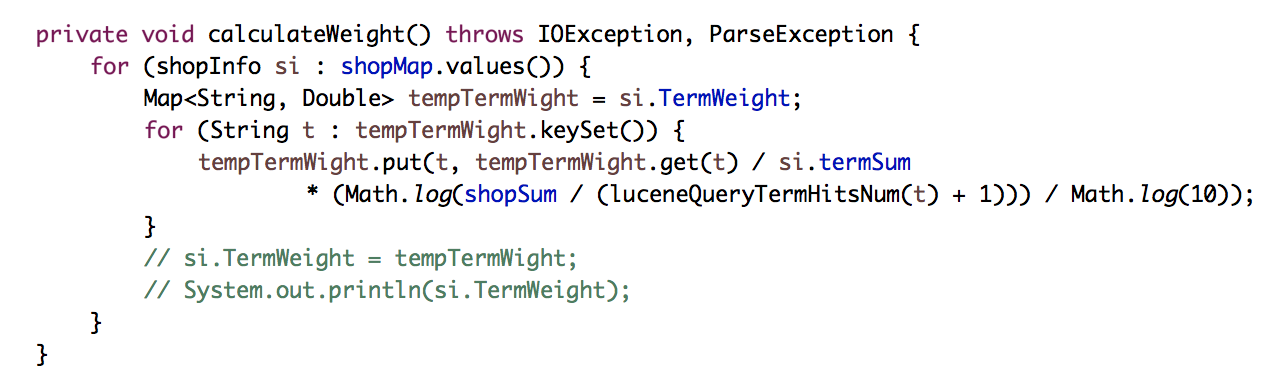
**2.文本向量化模块:** 文本向量化模块根据每个文本的分词结果计算出每个词项的tf-idf值，以向量形式表示文本。首先，统计出一个词项词典d，该词典包含数据集D全部的词项t。然后，统计出每个词项对应tf和df值。根据公式（2-1）和（2-2）tf-idf*t,d* = tf*t,d* × idf*t*计算出tf-idf值。至此，文档*p.φ*可以向量化表示为(w1, w2，···，wi)，。

文本检索中，我们需要根据关键词定位文档信息，所以我们需要一个从词项反向映射到文档的一种数据结构—反向文件。反向文件包含词项词典d，d中每一个词项都有一个记录包含该词项的所有文档的列表（称为倒排记录表，posting list），该表中的每一个元素表示词项在文档中的权重，即反向文件 = ((p1,w1),(p2,w2),...,(pi,wi))。表2-1为反向文件的一个示例：

表 2.1: 反向文件示例



文本向量化代码如下：



**3.空间文本索引构建模块**的任务分为构建 R 树和构建 IR 树两个主要部分。反向文件构建如下：



* + 1. **查询模块**

top-k 空间文本簇 (k-STC) 查询 q =< λ, ψ, k, ϵ, minpts > 有五个参数，查询模块会将参数分为 3 个字段，一个是文本字段，即一组关键词 ψ；一个是空间字段，即查询位置 λ；最后一个是调节参数，包含距离约束 ϵ，密度阙值 minpts，请求簇的数量 k。通过全文检索系统 Lucene 检索文档集，返回包含关键词的文档集 Dψ。选取一个文档作为核心对象执行聚类算法，找到以该核心对象为中心的簇，如果有满足参数条件 ϵ，minpts 的簇则计算簇得分加入候选列表。最终返回 k 个得分较好的簇。查询的BASIC算法实现代码如下：

