**Full-text Search**

结构化数据：指具有固定格式或有限长度的数据，如数据库，元数据等。

非结构化数据：指不定长或无固定格式的数据，如邮件，word文档等。

**基本思路**：

将非结构化数据中的一部分信息提取出来，重新组织，使其变得有一定结构，然后对此有一定结构的数据进行搜索，从而实现相对较快的搜索。这部分从非结构化数据中提取出的然后重新组织的信息，我们称之索引。

**反向索引**：搜索关键字(term) -> 文档列表(posting list)

"lucene" -> [3,6,9]

"solr" - > [1,2,4,5,7]

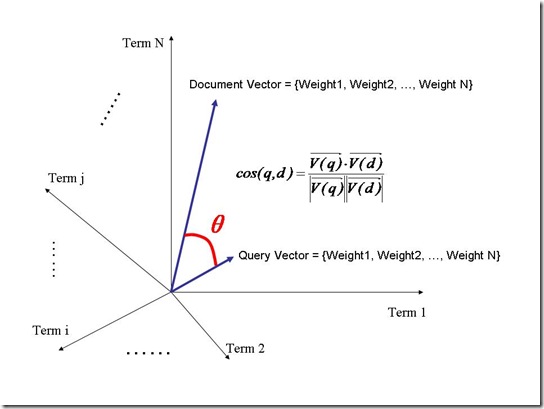
"hadoop" -> [6,8]

一次索引（慢），多次搜索（快）

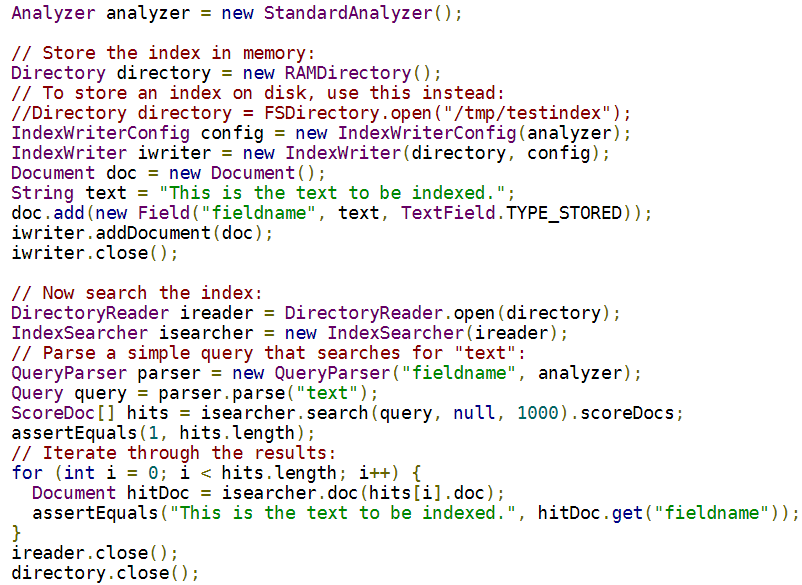
**相关性打分**：（以Vector Space Model为例）

Term Frequency (tf)：此Term在此文档中出现了多少次。tf 越大说明越重要，说明是中心内容。

Document Frequency (df)：有多少文档包含次Term。df 越大说明越不重要，说明不具有区分性。

weightt,d = tft,d · log(n/dft)

**Lucene, an indexing and search library, accepts only plain text input.**



To use Lucene, an application should:

1. Create Documents by adding Fields;
2. Create an IndexWriter and add documents to it with addDocument();
3. Call QueryParser.parse() to build a query from a string;
4. Create an IndexSearcher and pass the query to its search() method.

正向信息：索引(Index) –> 文档(Document) –> 域(Field) –> 词(Term)

反向信息：词(Term) –> 文档(Document)

Index File Formats

- An index contains a sequence of documents.

- A document is a sequence of fields.

- A field is a named sequence of terms.【term/token是搜索匹配的连接点】

- A term is a sequence of bytes.

**索引流程**

1. Parsing转换，形成Document

 Lucene does not care about the Parsing of these and other document formats, and it is the responsibility of the application using Lucene to use an appropriate Parser to convert the original format into plain text before passing that plain text to Lucene.

A Document is a collection of IndexableFields. A IndexableField is a logical representation of a user's content that needs to be indexed or stored.IndexableFields have a number of properties that tell Lucene how to treat the content (like indexed, tokenized, stored, etc.)

2. Tokenization分词，拆分得到Term

Tokenization is the process of breaking input text into small indexing elements – tokens. The way input text is broken into tokens heavily influences how people will then be able to search for that text.

3. Core Analysis分析，语义处理Term，生成Posting List

The analysis package provides the mechanism to convert Strings and Readers into tokens that can be indexed by Lucene.

**搜索流程**

1. 输入查询条件 -> 分词、运算符、生成Query查询树

2. 字符串匹配词元(term)、计算权重(term weight)、相关性打分(score)

3. Collectors are primarily meant to be used to gather raw results from a search, and implement sorting or custom result filtering, collation, etc. 收集器：收集并过滤搜索结果