考虑连接分布的依存语言模型设计和实现

1 得分函数

$$Score(Q,D)=(1-\phi)\sum_{i=1}^{n}P(q_i\mid Q)\log P(q_i\mid D)$$

$$+\phi\sum_{k=1}^{m}P(R_k\mid Q)\log P(R_k\mid D)$$

$$P(R_k\mid D)=(1-\eta)\frac{Fre_D(X)}{(X-1)(|D|-X/2)}$$

$$+\eta\frac{\alpha(L_X)\sum_{D_i\in C}Fre_{D_i}(X)}{(X-1)(|C|-yX/2)}$$

2 Lemur 概述

2.1 Lemur 工作流程

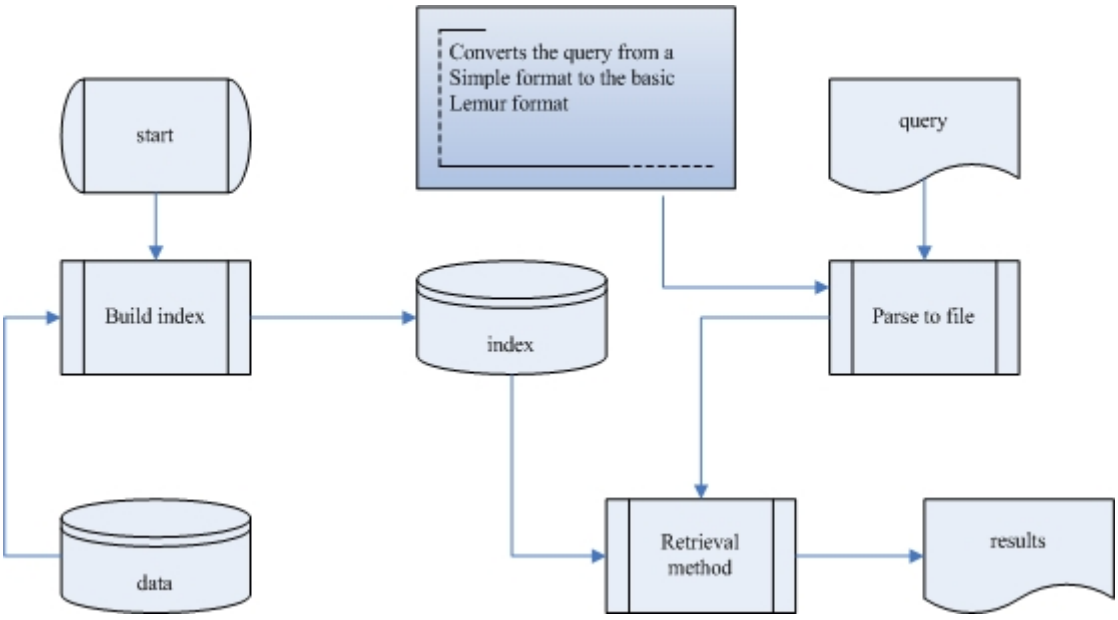
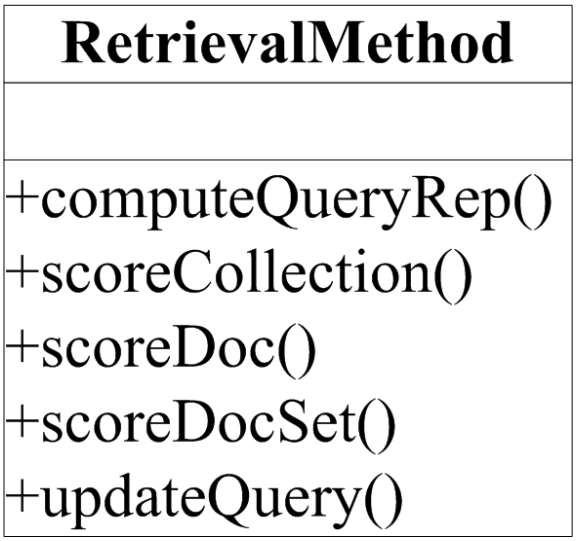


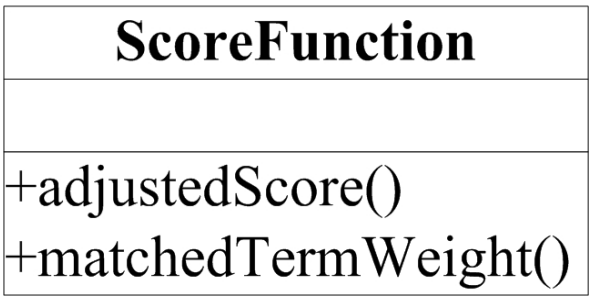
图 2.1：Lemur 工作流程图

2.2 Lemur 中实现检索模型的接口

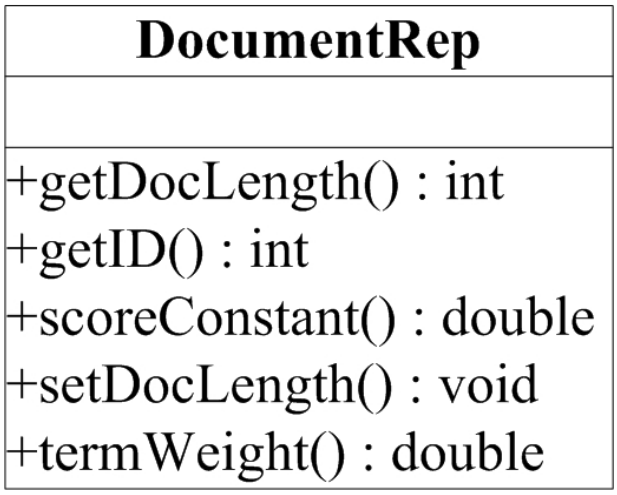
2.2.1 RetreivalMethod 类



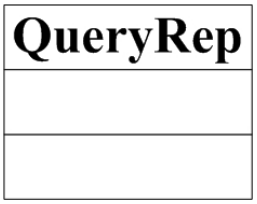
2.2.2 ScoreFunction 类



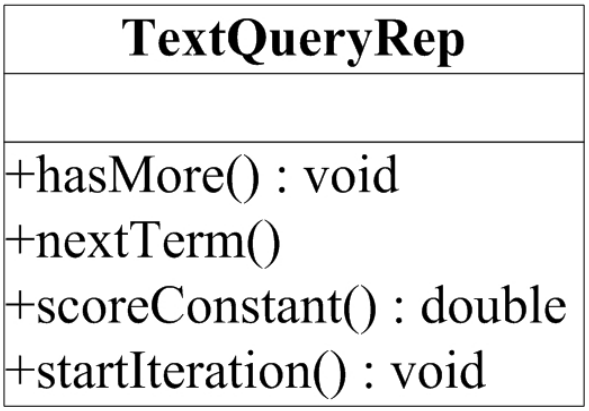
2.2.3 DocumentRep 类



2.2.4 QueryRep 类



2.2.5 TextQueryRep



2.3 相关 API

Index::term

term(char* s) : convert term string to a number

term(int id) : convert term number to a string

Index::document

document(char* s) : convert doc string to a number

document(int id) : convert doc number to a string

Index::termCount

termCount() : Total number of terms indexed

termCount(int id) : Total number of occurrences of term number id.

Index::documentCount

docCount() : Number of documents indexed

docCount(int id) : Number of documents that contain term number id.

Index::docLength(int docID)

The length, in number of terms, of document number docID.

Index::docLengthAvg

Average indexed document length

Index::termCountUnique

Size of the index vocabulary

Index::docInfoList(int termID)

Returns an iterator to the inverted list for termID.The list contains all documents that contain termID, including the positions where termID occurs.

Index::termInfoList(int docID)

Returns an iterator to the direct list for docID.The list contains term numbers for every term contained in document docID, and the number of times each word occurs. (use termInfoListSeq to get word positions)

3 考虑连接分布的依存语言模型工作流程

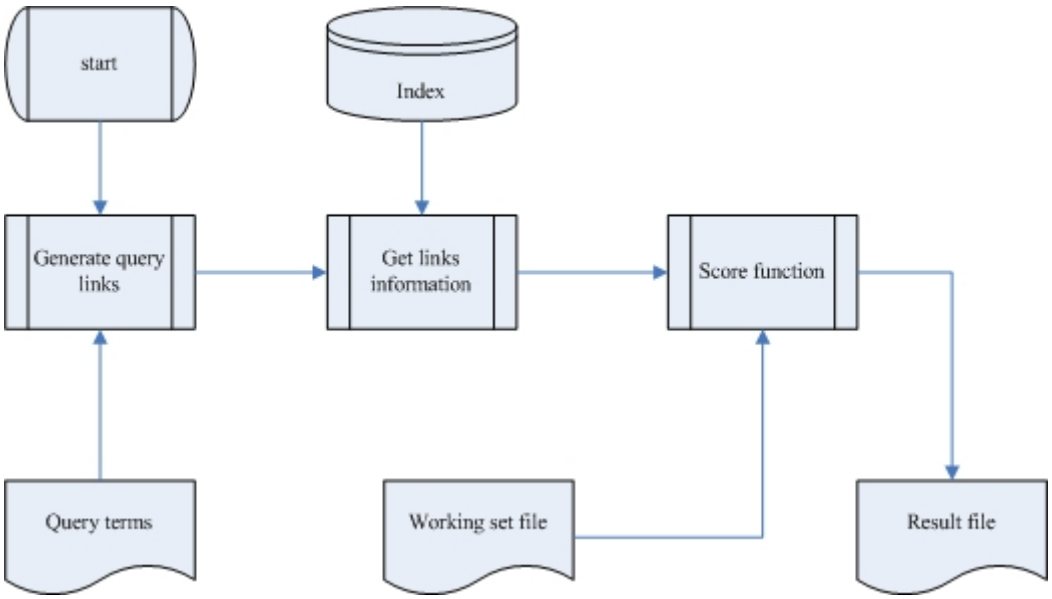


图 3.1 考虑连接分布的依存语言模型工作流程图

4 考虑连接分布的依存语言模型类关系图

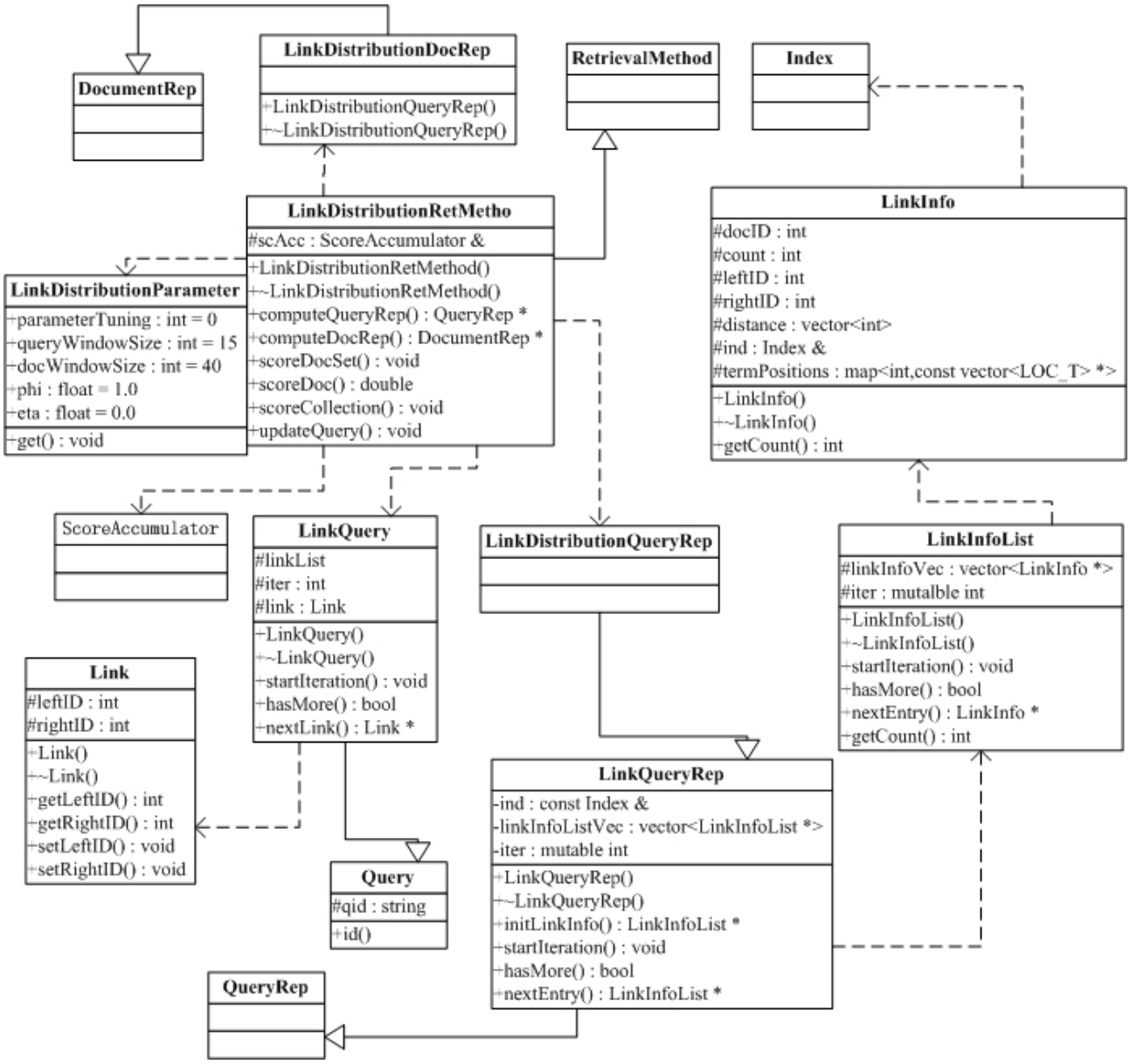


图 4.1 考虑连接分布的依存语言模型类关系图