Mestrado integrado em Engenharia de Computadores e Telemática

Departamento de Electrónica, Telecomunicações e informática

2015/2016

Recuperação de Informação

GRUPO 04

Autores:

- Bernardo Ferreira (67413)
- Bruno Silva (68535)

Responsável:

- Prof. Sérgio Matos

03/10/2016

Índice

1. Descrição das Classes
    1.1. Classes Principais
        1.1.1. Corpus Reader
        1.1.2. Document Processor
        1.1.3. Tokenizer
        1.1.4. Indexer
        1.1.5. Searcher
    1.2. Classes Auxiliares
        1.2.1. CorpusFile
        1.2.2. Document
        1.2.3. Index
        1.2.4. Posting
        1.2.5. Dictionary
2. Data Flow
3. Anexos

1. Descrição das Classes

Nesta Secção do documento vai ser realizada a descrição das classes usadas na implementação do motor de recuperação de informação. Esta descrição está sujeita a alterações no momento de implementação.

1.1. Classes Principais

1.1.1. Corpus Reader

O corpus reader é a classe responsável por percorrer um diretório recursivamente e devolver a lista de todos ficheiros e sua extensão, existentes em todo o diretório e respectivos subdiretórios.

Esta classe conta conta com os seguintes atributos:

- String[] ignoreExtensions - array que contém extensões de documentos que queiramos ignorar no processo.
- ArrayList<CorpusFiles> files - Array list que contém cada ficheiro e respetiva extensão presente no diretório.

Os métodos para esta classe são os seguintes

- readDir(String path) - método que lê o diretório fornecido no argumento e devolve um array de ficheiros com todos nos diretórios e subdiretórios.
- ArrayList<CorpusFile> getFiles() - Retorna lista de todos os ficheiros.
- CorpusFile getNextFile() - Retorna o próximo ficheiro, ou null caso não existam mais.

1.1.2. Document Processor

O document processor é a classe responsável por processar o documento consoante a sua extensão, retirar tags específicas do formato e criar um documento guardando o seu conteúdo num formato único em disco.

Esta classe conta conta com os seguintes atributos:

- String tmpPath - Localização do diretório temporário onde cada documento será guardado.
- ArrayList<Documents> documents - Lista de todos os documentos presentes nos ficheiros válidos encontrados pelo CorpusReader.

Os métodos para esta classe são os seguintes

- processDocument(CorpusFile cfile) - método que vê a extensão do ficheiro recebido e envia para a função interna de processamento adequada a sua extensão.

- processArrf(String path) - método que percorre o ficheiro do path recebido e cria um Documento para cada um encontrado dentro do ficheiro, procedendo ainda ao seu processamento(retirar tags etc.) e gravação do conteúdo do documento, num formato único, em disco.
- writeFile(String Document) - Função responsável pela gravação do conteúdo do documento, num formato único, em disco.

1.1.3. Tokenizer

O tokenizer é a classe responsável por ler o conteúdo do Documento produzido e aplicar transformações para se obter uma lista de palavras cujo o resultado será as formas simples de cada palavra(passar para letra minúscula, retirar pontuação, aplicar transformações linguísticas etc.)

Os métodos para esta classe são os seguintes:

- tokenize(Document doc) - método que lê o documento, aplica as transformações(chama a função transform) e escreve para o ficheiro tokenizado.
- String transform(String doc) - método que converte tudo para minúsculas, steamming, stopword filtering etc.

1.1.4. Indexer

O Indexer é a classe responsável por indexar cada palavra presente nos documento fornecidos.

Esta classe conta conta com os seguintes atributos:

- Index index - Estrutura de dados para suportar o Índice que vai ser gerado a partir dos documentos.

Os métodos para esta classe são os seguintes

- indexDocument(Document doc) - método para indexar um documento.

1.1.5. Searcher

O Searcher é a classe responsável pela pesquisa no Index por termos ou expressões e retorna quais os documentos onde as mesmas se encontram.

Esta classe conta conta com os seguintes atributos:

- Index index - Estrutura de dados que suporta o Índice que foi gerado a partir dos documentos.

Os métodos para esta classe são os seguintes

- Dictionary searchTerm() - método para pesquisar por um termo que retorna um Dictionary.

1.2. Classes Auxiliares

1.2.1. CorpusFile

O CorpusFile é a classe responsável por guardar informação de cada ficheiro encontrado pelo CorpusReader.

Esta classe conta conta com os seguintes atributos:

- String path - Path do ficheiro encontrado.
- String extension - Extensão do ficheiro encontrado.

Esta classe contém getters para todos os seus atributos.

1.2.2. Document

O Document é a classe responsável por guardar toda a informação relativa a documento(localização, id, etc).

Esta classe conta conta com os seguintes atributos:

- static String tmpPath - Path temporário para a localização do ficheiro com o conteúdo do documento.
- static Int id - contador do número de documentos.
- Int docId - Id do documento.
- Int docStartLine - Linha em que o documento começa dentro do ficheiro.
- String filePath - Localização do ficheiro onde se encontra o documento.
- String documentPath - Localização do ficheiro que contém apenas este documento.

Esta classe contém getters para todos os seus atributos.

1.2.3. Index

O Index é a classe responsável pela estrutura de dados usada para a indexação de palavras presentes nos documentos.

Esta classe conta conta com os seguintes atributos:

- HashMap<Integer, Dictionary> - Estrutura de dados do Índice.

Os métodos para esta classe são os seguintes:

- addTerm(String term, int doc) - método para adicionar um termo no índice em que aparece no documento. Se já existir, não cria uma entrada nova, mas sim, adiciona o documento à Posting List.

- Dictionary searchTerm(String term) - método para pesquisar por um termo que retorna um Dictionary.
- removeTerm(String term) - método para remover um termo do Índice.
- removeDocument(String Term, int doc) - método para remover um documento da posting list respetiva daquele termo.

1.2.4. Posting

Classe que guarda a lista dos documentos em que um termo aparece.

Esta classe conta conta com os seguintes atributos:

- ArrayList<Integer> docs - Estrutura de dados do Posting.

Os métodos para esta classe são os seguintes:

- addPosting(int doc) - método para adicionar um novo posting.
- int postingListSize() - método para retornar o tamanho da lista dos postings.
- removePosting(int doc) - método para remover um posting.
- ArrayList getAllPostings() - método que retorna a lista de todos os postings.

Esta classe contém getters para todos os seus atributos.

1.2.5. Dictionary

Classe que guarda a informação do Índice, por exemplo, os termos, o número de documentos em que o termo aparece e um ponteiro para a lista de Postings.

Esta classe conta conta com os seguintes atributos:

- String term - Termo presente no dicionário.
- int nDocs - Número de documentos onde um determinado termo aparece.
- Posting postingList - Posting onde é guardada a lista dos documentos em que um termo aparece.

Os métodos para esta classe são os seguintes:

- addDocument(int doc) - método para adicionar um documento à posting list.
- removeDocument(int doc) - método para remover um documento à posting list.
- Dictionary getTerm() - método para retornar informações sobre o próprio termo.

2. Data Flow

O nosso Information Retrieval Engine, contém uma main que é responsável por instanciar as classes principais referidas acima.

A indexação de um corpus inicia-se no CorpusReader que fornece uma lista de ficheiros contendo documentos para serem indexados. Essa lista é posteriormente consumida pelo DocumentProcessor que irá dividir os documentos presentes nos ficheiros num formato único, retirando características específicas do formato do ficheiro.

A terceira fase do processo consiste no Tokenizer fazer o tratamento de cada documento, transformando-o numa lista de tokens, que de seguida será utilizada pelo Indexer. Para o processo de indexação, o Indexer percorre todos os tokens de cada documento e preenche uma estrutura de dados(Index) que é dependente da implementação pretendida.

Por fim para se efetuar pesquisas sobre os documentos indexados, é utilizado o Searcher que usando a mesma estrutura de dados(Index), devolve uma lista de documentos, ordenados ou não dependendo da pesquisa avaliar os resultados por rank, onde o termo se encontra.

3. Anexos

Por motivos de visualização, o diagrama de classes encontra-se num ficheiro à parte.