Apache Lucene

Marcos Felipe Eipper[1], **Willian Feldmann Kumlehn**[2]

[1]Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina (UDESC/CCT)
Joinville – SC – Brazil

marcos@eipper.com.br, willianfkumlehn@gmail.com

Abstract. *This article introduces some aspects and applications of the Apache Lucene platform. It contains the definitions of Stemming in english and portuguese, as well as alternatives to improve its algorithm. It also explicits the Lucene's architecture, introducing the indexing structure, it's efficiency, document removal and how to adjust it's performance.*

Resumo. *Este artigo apresenta alguns aspectos do funcionamento e aplicação do Apache Lucene. Nele está contido as definições de Stemming, inglês e português, bem como alternativas pero melhorar o algoritmo deste último. Também explicita a arquitetura do Lucene, apresentando a estrutura de índices, a eficiência destes, a remoção de documentos e como ajustar a performance do programa.*

1. Introdução

O Apache Lucene é um software de busca escrito em Java, open-source, da Apache Software Foundation. Se trata também de um framework poderoso e flexível de indexação de documentos. O Lucene pode tratar diversos tipos de dados, desde que possam ser convertidos para texto. O ElasticSearch tem como núcleo de seu motor de buscas o Lucene, assim como a Wikipedia em seu sistema de busca de artigos.

2. Funcionamento

Para que uma busca seja feita de forma eficiente, o Lucene se baseia na estrutura de dados chamada índice. O índice permite que um termo seja localizado de forma rápida, bem como onde dentro do conteúdo os termos podem ser encontrados.

Índices são estruturas complexas, e criá-los demanda muito de um computador. Portanto, dividimos a aplicação em dois grupos: indexação e consulta. Podemos analisar o funcionamento do Lucene imaginando um grande tubo, com uma estrutura em seu centro.

2.1. Do lado esquerdo do tubo: construção do índice

2.1.1. Entrada

Temos uma entrada de conteúdo bem no início do tubo. Assim que um documento entra, a aplicação deve adquirir o conteúdo, para então entrar onde o Lucene age.

2.1.2. Dentro do Lucene

Primeiramente o Lucene deve construir seu tipo predefinido de documento. Para poder novas entradas no índice (i. e., documentos), o Lucene utiliza um IndexWriter (escritor de índices), que recebe como parâmetros: o arquivo de diretório do índice e um Analyzer.

Então, passamos para a etapa de análise. A ocupação do Analyzer é de ”traduzir”cada campo dos dados em ”tokens”ou palavras-chave.

2.2. Centro

Passada a etapa de análise, temos que adicionar os documentos ao índice. Cada documento de índice possui campos de identificação, como por exemplo: ”nome do arquivo”, ”caminho para o arquivo no sistema de arquivos”, ”conteudo do arquivo”. Nesta etapa o Lucene utiliza o IndexWriter que foi criado na primeira etapa para efetivamente gravar o índice ao disco.

Dentro de nosso tubo imaginário, aqui fica a grande estrutura chamada índice.

2.3. Do lado direito do tubo: interface com o usuário

2.3.1. Construção da consulta e Processamento de Resultados

Na entrada mais à direita de nosso tubo imaginário, temos **entrada** e **saída**. Agora que os dados estão devidamente indexados, podemos fazer consultas. Na maioria dos casos, serão necessárias duas classes para suportar buscas completas de texto: QueryParser e IndexSearcher. Acoplando ambas as classes dentro de uma nova chamada, por exemplo, MotorDeBusca, podemos criar métodos como por exemplo:

fazBusca(String termoASerBuscado); – QueryParser para traduzir a busca e passa o objeto Query ao IndexSearcher.

pegaDocumento(int docId); – retorna do IndexSearcher o documento encontrado.

3. Stemming

Stemming Stemming é uma técnica de redução de palavras em sua forma flexionada ou derivada à sua base, utilizado em algoritmos e funções de busca de informação. O processo de redução se baseia na transformação de palavra em suas predecessoras, ou seja, uma raiz ou base, a qual seja conhecido e apresentada em uma tabela para se otimizar e facilitar o processo de busca.

Stemming Português Assim como descrito acima, o processo de stemming para língua portuguesa funciona através da redução das palavras derivadas. Como a língua portugues possui uma morfologia razoavelmente complexa, com diversas regras e tipos de afixação, o algoritmo utilizado para o processo é razoavelmente maior que o da língua inglesa e mais complexo. Utiliza para seu funcionamento um algoritmo de remoção de sufixo e prefixos, um tamanho mínimo para a base, raiz, da palavra e trata as excessões de forma separada, através, geralmente, de uma lista.

Stemming Inglês A forma inglesa da técnica de processo de stemização funciona de forma similar à portuguesa, reduzindo as palavras a sua ”root”, raiz, e se diferencia pelos algoritmos e forma de redução, uma vez que a gramática inglesa é bastante diferente

da portuguesa. Como a língua ingluesa possui uma morfologia bastante simples, há uma facilitação do processo. De forma geral o processo de stemming funciona utilizando-se uma ”lookup table”onde se armazenam as palavras derivadas e é feita uma consulta à elas, as quais ficam associadas a suas palavras raízes. Logo, com uma morfologia mais simples, a tabela inglesa é construída com algoritmos menores e não possui um tamanho tão grande. Alguns algorirmos comuns utilizados no processo de redução da língua inglesa são os de remocão de prefixos ou sufixos, ou ainda afixos, que tratam ambos os casos. O de sufixo, por exemplo, trabalha com algumas regras básicas da língua, removendo fins comuns de sufixação, os quais são ”ed”, ”ing”e ”ly”. Apesar da língua ter diversas exceções, elas ainda podem ser facilmente tratadas e há um grande ganho de tempo processacional.