Trabalho Prático III - Buscador

Thiago Vieira de Alcantara Silva
2012075627

1 Introdução

Neste trabalho, desenvolvi um processador de consultas usando um modelo vetorial. Dado um conjunto de dados e uma consulta, o processador retira os termos da consulta formando um vetor, então calculamos a similaridade da consulta com cada documento. Os documentos mais similares são considerados como resposta.
Para a execução das consultas desenvolvemos uma página html onde o usuário pode inserir suas consultas e obter os documentos mais similares a ela.

2 Decisões de Projeto

2.1 Estrutura do Projeto

O processador de consultas foi desenvolvido em C++. A pasta principal do projeto é dividida em seis pastas diferentes:

1. base: Nesta pasta temos todas as classes que são usadas pelos buscadores implementados no TPII ou TPIII.

2. boolean: Nesta pasta temos o buscador implementado para o segundo trabalho prático.

3. doc: Na pasta doc guardamos a descrição e a documentação do trabalho.

4. search: Em search podemos encontrar as páginas html que podem ser usadas para executar consultas.

5. util: Na pasta util temos scripts utilitários que servem para executar tarefas específicas.

6. vectormodel: Nesta pasta temos o buscador implementado para o terceiro trabalho prático. Este buscador recupera 20 documentos e os ordena de acordo com a similaridade com a consulta executada.

2.2 Arquivos Auxiliares, Dependências e Execução

O processador em vectormodel depende somente das classes da base.
Para compilar o processador de consultas basta executar o utilitário Make que se encontra dentro da pasta vectormodel. Existe uma opção de execução disponível, podemos informar o nome do arquivo em que queremos salvar a resposta do processador com a option -o. Assim se executarmos ./main -o [Nome do arquivo] o resultado da consulta será salvo em um arquivo.
Na pasta vectormodel temos um arquivo chamado CONFIG, nele guardamos o nome do arquivo que contém o índice, o nome do arquivo que contém a lista de urls e o nome do arquivo que contém o vocabulário da base de dados indexada. Ao executar o binário do processador de consultas, a primeira coisa que ele faz é ler o arquivo CONFIG e carregar os arquivos descritos nele.

3 Ranking

A escolha e rankeamento dos arquivos que devem ser retornados após uma consulta é feita usando um cálculo de similaridade entre o vetor da consulta e os documentos da base de dados indexada. O cálculo de similaridade feito neste trabalho foi o cosseno. Um vetor de um documento ou consulta pode ser enxergado como um vetor n-dimensional onde cada dimensão representa uma palavra do vocabulário, a magnitude do valor de cada dimensão do vetor é calculada com base na frequência que cada termo aparece no documento. Assim o documento mais relevante de acordo com o processador será o que tiver o maior valor de cosseno com a consulta.

4 Análise de Complexidade

Buscas são feitas percorrendo o índice invertido e guardando o peso de cada documento em memória principal, assim para cada 1 milhão de páginas na base de dados, gastaremos pouco mais de 1 Mbytes de dados por consulta executada.

Em relação ao tempo, como para cada tripla do índice atualizamos o valor do vetor dos documentos, a complexidade assintótica de uma busca é

O(I * log n) onde I é o número de triplas do índice invertido e n é o número de documentos da base de dados.