Trabalho Prático de Recuperação de Informação - 2015.2

Anderson Pimentel dos Santos[1]

[1]Universidade Federal do Amazonas - UFAM

Matrícula 2150260

1. Informações Gerais

Trabalho prático para disciplina de Recuperação de Informação(RI) do Instituto de Computação(ICOMP). O objetivo do trabalho é a implementação de uma máquina de busca para a coleção de documentos CFC(Cystic Fibrosis Collection), utilizando o *Modelo Vetorial* como modelo de similaridade para as tarefas de RI. A coleção CFC foi disponibilizada no endereço eletrônico: http://coyote.icomp.ufam.edu.br/ir/cfc.tar.gz

2. Descrição de Implementação

O trabalho foi implementado na linguagem C++, com algumas estruturas encontradas na versão 2011(C++11). Abaixo estão as estruturas utilizadas em geral:

unordered_map Estrutura de hash do c++11 utilizada para armazenar o vocabulário, norma dos documentos e acumulador de similaridades para documentos;

document Tipo abstrato de dados(TAD) representando os documentos da coleção e consultas;

parser e qparser Componentes utilizados para ler a coleção de documentos e consultas e retornar o vetor de documentos(*document*);

Hash_Table TAD que representa o índice invertido da coleção e faz uso da estrutura *unordered_map* para armazenar os termos(chaves) e suas listas invertidas;

Term e Doc Representam o termo com seu idf e frequencia; Representa a lista encadeada de documentos que possuem tal termo e sua frequencia;

Iwriter e Ireader Componente de escrita e leitura do índice invertido utilizado pelo indexador para indexar a coleção e ser recuperado para memória pelo processador de consultas;

indexer e qprocessor Indexador da coleção e processador de consultas;

Query e Score Representa a consulta e conteúdo; Representa a combinação do documento e sua similaridade quanto à consulta submetida;

util Funções variadas servindo de auxílio para componentes e TADs.

2.1. Detalhes de Implementação

O trabalho de implementação pode ser divido em duas fases: Indexação e Processamento de Consultas. No primeiro, a estratégia de implementação foi de indexar a base em dois arquivos: *base.ref* que guarda as informações de referência do início da *struct* que representa o termo e sua lista invertida; *base.dat* que armazena tadas as *structs* do índice invertido. Foi gerado também o arquivo *norma.ref*, que contem a norma de todos os documentos preprocessados no indexador.

Na segunda fase, o processador de consultas extrai todas as consultas e carrega para memória o arquivo *base.ref*, e com o componente *Ireader*, carrega na memória as estruturas necessárias para a consulta. O cálculo de similaridade e criação do *ranking* são realizados apenas calculando o peso dos elementos da consulta e utilizando as informações preprocessadas do termo da coleção e norma.

3. Resultados

Os resultado nas 100 consultas foram de 49,60% para P@10 e 30,07% para MAP. O tempo de processamento para as consultas obteve média de 3 milisegundos por consulta. Informações detalhadas podem ser obtidas no arquivo *result.txt* gerado após o processamento das consultas;

3.1. Estudo dos Resultados

Durante a implementação, a decisão de utilizar o valor T_f como normalização logarítmica, ao invés da frequência bruta do termo, em conjunto com a remoção de palavras menores que 3 caracteres na indexação melhorou em 1% o resultado do P@10 e diminuiu em 0,06% o resultado de MAP, para as 100 consultas. Por fim, foi usado o peso de 0,05 para o campo de palavras-chave de *Minor Subject*, em conjunto com a indexação do mesmo e do campo de *Major Subjects*, o que melhorou o resultados do P@10 em 1,4% e do MAP em 1,35% para as 100 consultas.

4. Endereço Eletrônico da Implementação

A implementação desse trabalho pode ser obtida no endereço eletrônico: https://github.com/andbrain/cfc

5. Como Usar

Para obter os resultados descritos nesse trabalho, os passos abaixo devem ser realizados.

Na pasta raiz onde o trabalho foi clonado ou descompactado, use os comandos no terminal:

1. *make init* – criação das pastas para inicialização
2. Colocar os arquivos na pasta *base* criada na pasta raiz. Obs: Os arquivos da coleção devem seguir os padrão *cf74, cf75, . . . , cf79* e o arquivo de consulta deve ter o nome *cfquery*
3. *make* – compilação do código-fonte
4. *./indexer* – indexação da base
5. *./engine* – processamento das 100 consultas

6. Arquivos Gerados

Após o processo, são criados os arquivos:

result.txt Resultados detalhados do processamento das 100 consultas, com métricas e tempo de processamento;

ranking.txt Ranking de documentos com sua similaridade para as 100 consultas;

base.ref Arquivo de texto usado para refenciar as estruturas da coleção;

base.dat Arquivo binário com todas as estruturas da coleção;

norma.ref Arquivo de texto com a norma de todos os documentos;

vocabulary.txt Arquivo de texto com informações do índice invertido usado para indexar a base;