Trabalho Prático de Recuperação de Informação
Implementação do Modelo Vetorial

Josiane Rodrigues da Silva[1], **Matrícula: 3150044**

[1]Universidade Federal do Amazonas - ICOMP/UFAM

josiane@icomp.ufam.edu.br

1. Objetivo

O desenvolvimento desse trabalho consiste em implementar uma máquina de busca para uma coleção de documentos. O modelo de Recuperação de Informação implementado foi o Modelo Vetorial. Como já descrito na literatura, o Modelo Vetorial é um modelo clássico e bastante utilizado em Recuperação de Informação.

A coleção de documentos utilizada foi a CFC (Cystic Fibrosis Collection). Essa coleção é composta por 1.239 documentos publicados entre 1974 e 1979 sobre a doença genética Fibrose Cística, e é formada por 6 arquivos de documentos e um arquivo contendo 100 consultas e suas respectivas respostas relevantes.

Dessa forma, o objetivo desse trabalho é apresentar a implementação do Modelo Vetorial com o intuito de processar as 100 consultas da coleção, retornando um ranking de respostas, em função dos Record Numbers dos artigos de resposta, para cada consulta.

Para avaliação dos resultados, realizou-se a comparação dos resultados retornados pelo sistema com a base de relevantes da coleção, utilizando as métricas MAP e P@10, onde foram considerados como relevante todos os documentos citados no arquivo de consultas, sem considerar a nota de relevância dos avaliadores.

2. Implementação

Para a implementação desse trabalho foi utilizada a linguagem de programação C++. As bibliotecas utilizadas nesse trabalho foram:

- `iostream`: para manipulação de fluxo de dados padrão do sistema (entrada padrão, saída padrão e saída de erros padrão);
- `fstream`: para manipulação de fluxos de dados de arquivos de computador;
- `string`: para manipulação de cadeias de caracteres;
- `vector, map, unordered_map`: estruturas de dados;
- `sstream`: para permitir o uso da função `copy()`, usada nesse trabalho para separar os termos dos documentos e da consulta;
- `iterator`: para percorrer estruturas de dados como vetores e maps;
- `algorithm`: para o uso da função `remove_if`;
- `ctype`: para o uso da função `is_digit()` para verificar se um dado caracter é um dígito;
- `cctype`: para o uso da função `tolower()` para colocar as palavras em caixa baixa;
- `math.h`: para o uso da função `log2()`.

Para criação do índice invertido e processamento da consulta foram utilizadas principalmente as estruturas map (por possibilitar a associação de chave a um valor) e vector. Para tanto foram definidas três estruturas (usando a definição de struct): uma estrutura para a lista invertida, armazenando o tf e o peso do termo; outra para o índice invertido para armazenar o total de documentos que contém o termo, o idf e sua respectiva lista invertida; e finalmente uma estrutura para armazenar valores de cada consulta, tais como: tf, peso, norma, entre outros.

Os termos indexados da coleção foram: “TI”, “AB”, “EX”, “MJ”e “MN”.

3. Link do GitHub

A implementação desse trabalho está disponível em: https://github.com/josianerodrigues/Trabalho_RI

4. Compilação e execução da implementação

4.1. Organização dos arquivos

O diretório disponível no GitHub contém:

- o diretório `src/`: que contém todos os arquivos com os códigos fonte.
- `stopwords.txt`: arquivo que contém a lista de *stopwords* que devem ser removidas durante a criação do índice invertido e do processamento da consulta.
- `relatorio.pdf`: relatório do trabalho.

4.2. Como compilar e executar

Para compilar este trabalho entre no diretório `src/` e execute o comando `make`. E para executar faça: `./out > [nome_arquivo_de_saída.txt]`. O resultado será gerado no arquivo de saída definido. Lembrando que o diretório com os arquivos da coleção devem ser colocados no diretório raiz desse trabalho. Além disso, é gerado um arquivo chamado `hash.txt` contendo o índice invertido da coleção, tudo isso dentro do diretório `src/`.

5. Resultados

5.1. Métricas de Avaliação

Como mencionado anteriormente, para a comparação dos resultados retornados pela máquina implementada, foram utilizadas as métricas MAP interpolado e P@10 para a base de consultas realizadas. Para cada consulta foi calculado o MAP e o P@10 e no final foi tirado a média das métricas das 100 consultas. Foi obtido os seguintes resultados:

- Map interpolado: 28,25%
- P@10: 43,5%

Para a coleção CFC algumas consultas retornaram resultados muito baixos, o que pode ter prejudicado o resultado final.

5.2. Tempo de processamento

Para este trabalho foi utilizado o sistema operacional Linux 64-bit, processador intel Core i5 (1,60GHzx4), memória ram de 4GB e compilador G++. Nessas condições a tempo de execução do modelo para processar as 100 consultas é de aproximadamente 5 segundos (resultado mostrado no final da execução do trabalho).