DOCUMENTAÇÃO MÁQUINA DE BUSCA

RELEASE:1

DATA RELEASE: 13/06

-

Introdução

Uma máquina de busca são robôs capazes de recuperar a informação dentro de uma base de dados específica, dado uma consulta (query) feita por um usuário. O seu funcionamento é dado pela necessidade que o usuário tem de uma determinada informação. Ele então expressa essa necessidade para o sistema na maioria das vezes através de um conjunto de palavras-chave, que denominamos de consulta. Com a consulta montada o sistema de buscas vai até a base de dados de documentos e tenta recuperar as respostas que são mais relevantes para a consulta especificada.
O modelo de representação a ser utilizado neste trabalho é o modelo vetorial. Este tipo de sistema vê seus componentes (documentos e consultas) como um conjunto de palavras-chave, mais especificamente como vetores, onde é salvo o peso de cada termo (cada palavra do vocabulário). Em um modelo vetorial as coordenadas de um vetor qualquer são determinadas pelas palavras que o descrevem. Desta forma, temos que o número de palavras distintas da coleção determina a dimensão do espaço onde os documentos e consultas serão representados, ou seja, se o vocabulário tem n palavras distintas o número de dimensões é n (Rˆn).

Implementação

Indice_Invertido:

Dada a coleção de documentos, um índice invertido é uma estrutura contendo uma entrada para cada palavra (termo) que aparece em pelo menos um documento. Para cada palavra, o índice invertido armazena a lista de documentos que a contém.

Indice_Invertido.cpp

`File_reader::Ler()`	Função para ler os documentos da coleção.
`File_reader::Imprimir()`	Função para imprimir os documentos da coleção.
`File_reader::mapa(`	Função para retornar o índice.
`File_reader::pertence (string palavra)`	Retorna se a palavra existe até o end do map.
`File_reader::doc_number (string c)`	Retorna o tamanho do set da string
`int ocorrencias(string palavra_nova, string documento)`	Abre o documento e enquanto documento aberto, lê todas as palavras.
`int File_reader::doc_quantity()`	Retorna o número de documentos da coleção.

mapa.cpp:

Dicionários são estruturas associativas que armazenam elementos combinando uma chave com um valor. A chave é usada exclusivamente para identificar o elemento, que contém um valor mapeado.Neste caso, as chaves são todas as palavras contidas nos documentos e o valor associado a cada chave é o conjunto (set) com os nomes dos documentos.

mapa.cpp

`double dicionario::idf(string palavra`	Retorna a importância do Pt (palavra na coleção) do documento de acordo com o mapa do índice invertido.
`double dicionario::tf(string palavra)`	Retorna a frequência de palavras Pt no documento dj.
`double dicionario::w(string palavra`	Retorna a coordenada do documento dj no eixo Pt.
`void dicionario::cosine_ranking()`	Lê os arquivos e atribuiu a um map a string do documento e o valor do seu vector, e seu cosine ranking a um vector.
`map<string, vector<double>> map_return()`	Retorna o map mapa_busca
`double sim(string palavra)`	Realiza o cálculo de SIM para todos os documentos

cosine_ranking:

Cosine Ranking(Ranking Cosseno) que faz uso da estrutura que armazena as coordenadas dos documentos para rankear consultas utilizando uma base de dados textual.

mapateste.cpp:
Testes de unidade.

TEST_SUITE("File_reader")

`TEST_CASE("void Ler()")`	Testa a função void Ler()
`TEST_CASE("bool pertence(string palavra)")`	Testa a função bool pertence(string palavra)
`TEST_CASE("int doc_number(string c)")`	Testa a função int doc_number(string c)
`TEST_CASE("int doc_quantity()")`	Testa a função int doc_quantity()
`TEST_CASE("int ocorrencias(string palavra, string documento))")`	Testa a função int ocorrencias(string palavra, string documento))

Conclusão

Nós chegamos a conclusão que o trabalho foi bem sucedido e muito proveitoso, conseguimos aplicar grande parte da nossa matéria de PDS2 que aprendemos durante o semestre e, portanto, a máquina de busca fechou com chave de ouro todo esse aprendizado.
Além disso, realizamos os testes abrangendo uma grande quantidade de possibilidades que possibilitou um grande conhecimento dessa última matéria.