UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Departamento de Ciência da Computação

DCC207 -- Algoritmos 2
Prof. Renato Vimieiro

Trabalho Prático 1 – Manipulação de sequências

Objetivos

Nesse trabalho, serão abordados aspectos práticos dos algoritmos vistos em aula para manipulação de sequências. Especificamente, serão explorados aspectos de implementação de árvores de prefixo para construção de índices invertidos.

O objetivo secundário é fixar o conteúdo e mostra sua aplicabilidade em contextos práticos mais realistas. Entende-se que ao implementar a estrutura o aluno conseguirá compreender melhor os conceitos explorados. Dessa forma, o conteúdo teórico será melhor absorvido e fixado. Além disso, os alunos terão a oportunidade de ver conceitos não abordados na disciplina, no caso específico, bibliotecas para construção de aplicações web e conceitos relacionados a recuperação de informação e máquina de busca.

Tarefas

Nesse trabalho prático, os alunos deverão implementar um protótipo de máquina de busca. Serão implementados módulos para indexação de documentos, recuperação de documentos (consulta) e interface com o usuário. O módulo de coleta não será abordado. Os documentos serão restritos a um corpus do portal de notícias BBC News disponibilizados por Greene e Cunningham (2006). Esse corpus contém 2225 notícias de cinco áreas temáticas diferentes: economia, entretenimento, política, esportes e tecnologia. O link para o corpus está na seção de Links Úteis.

O módulo de indexação deverá criar um índice invertido dos documentos disponibilizados no corpus. O índice invertido associa uma palavra/termo aos documentos em que ela ocorre. Durante a execução do aplicativo, o índice deverá ser mantido em memória principal. A implementação deve ser feita através de uma árvore Trie compacta. A estrutura deverá ser inteiramente implementada pelos alunos. Deverá ser criado um módulo Python exclusivo para a implementação da estrutura. O módulo do índice invertido usará essa estrutura. O módulo deverá conter funções para indexação inicial dos documentos, isto é, caso não exista um índice construído, ela deve criá-lo em memória a partir do processamento dos documentos. Esse passo consiste na leitura dos documentos e associação dos termos aos documentos, os quais serão armazenados na Trie compacta. O módulo também deverá ter funções para o carregamento e armazenamento de um índice em disco. Isto é, caso exista um arquivo com o índice, este deverá ser carregado para a memória principal e, ao fim da

execução, ele deverá ser persistido em disco, caso ainda não exista. O armazenamento deverá ser feito em formato próprio. Não poderão ser usados arquivos pickle ou outro formato para armazenamento de objetos em disco. A escolha do tipo de arquivo, binário ou texto, fica a critério da implementação. No entanto, deve-se documentar tal escolha.

O módulo de recuperação de informação (RI) deverá conter funções para o processamento de consultas e obtenção dos resultados. Nesse trabalho, adotaremos um modelo híbrido para a máquina de busca. Nossa consulta será uma expressão Booleana formada pelos termos/palavras da busca e conectores lógicos para restrição dos resultados. Os conectores que poderão ser usados nas consultas são AND e OR (através dessas palavras-chave – a escrita deve ser toda em maiúscula para diferenciar de consultas envolvendo esses termos). Também podem ser usados parêntesis para mudar a precedência dos operadores. Assim, uma possível consulta seria "(casa AND piscina) OR praia".

O módulo de RI deverá analisar a consulta, recuperar os documentos que contenham os termos da pesquisa, e retornar a lista de resultados. A lista com os resultados deverá estar ordenada em ordem decrescente de relevância. A relevância de um documento para a consulta é a média dos z-scores dos termos da consulta (o quão mais frequente esses termos são no documento comparados com a média da frequência no corpus).

A exibição dos resultados deverá conter um trecho (snippet) do documento onde o termo ocorre. Esse trecho deverá ser composto por 80 caracteres que antecedem o termo mais relevante que ocorre no documento e 80 caracteres posteriores. O termo em si deverá ser destacado nesse trecho. A lista exibida deverá ser paginada, limitando 10 resultados por página. O módulo de RI deve ser capaz de interagir com o front-end para reportar os resultados corretamente.

Em relação ao front-end, além das funcionalidades listadas, serão avaliados aspectos de usabilidade do sistema e qualidade visual (aspectos gráficos do design) da solução. A implementação deverá ser feita obrigatoriamente em Flask.

Finalmente, os alunos deverão preparar um relatório final em que descrevem textualmente sua implementação bem como o problema abordado no trabalho. Deverão ser descritos todos os passos da implementação, descrevendo as decisões tomadas e a exibição de exemplos ilustrando o resultado ou mecanismo implementado. Também deverão ser dados exemplos de funcionamento do sistema. O nível de elaboração do texto e qualidade das descrições serão critérios de avaliação. Em outras palavras, o mesmo cuidado com a implementação deverá ser observado no relatório produzido. O relatório deverá ser publicado junto com o código em repositório aberto no GitHub.

O trabalho poderá ser feito em **grupos de até dois alunos**. Recomenda-se fortemente que o trabalho seja realizado em grupo.

A implementação deverá ser feita em Python 3.9+. O uso de qualquer biblioteca adicional ou código de terceiros deverá ser discutido com o professor. Essa restrição não se aplica à biblioteca Flask, restrito somente à implementação da interface e servidor da aplicação.

O que entregar?

Deverá ser entregue um repositório no GitHub contendo todos os arquivos criados na implementação da ferramenta. O link para o repositório deverá ser postado no Moodle. O repositório deverá ser mantido privado até a data final de entrega (após tolerância por atraso). Então, o repositório deverá ser tornado público. **Caso o repositório não esteja aberto ou o link postado não funcione a partir do 1º dia após a data final de entrega, o trabalho será considerado não entregue e receberá nota nula.**

Política de Plágio

Os alunos podem, e devem, discutir soluções sempre que necessário. Dito isso, há uma diferença bem grande entre implementação de soluções similares e cópia integral de ideias. Trabalhos copiados na íntegra ou em partes de outros alunos e/ou da internet serão prontamente anulados. **Caso haja dois trabalhos copiados por alunos/grupos diferentes, ambos serão anulados**.

Datas

Entrega: 26/10/2025 às 23h59

Política de atraso

Haverá tolerância de 30min na entrega dos trabalhos. Submissões feitas depois do intervalo de tolerância serão penalizados, incluindo mudanças no repositório.

- Atraso de 1 dia: 30%
- Atraso de 2 dias: 50%
- Atraso de 3+ dias: não aceito

Serão considerados atrasos de 1 dia aqueles feitos após as 0h30 do dia seguinte à entrega. A partir daí serão contados o número de dias passados da data de entrega.

Links úteis

- https://flask.palletsprojects.com/en/stable
- http://mlg.ucd.ie/files/datasets/bbc-fulltext.zip