TP3 – Índice invertido

O terceiro trabalho prático da disciplina também envolverá implementar as buscas por produtos por meio de palavras (termos) do nome destes produtos.

ÍNDICE INVERTIDO:

O índice invertido nos permite fazer buscas por entidades a partir de seus termos (palavras). Para isso, a gente precisa criar uma lista de IDs para cada termo indexado. Por exemplo, suponha que os seguintes produtos estejam cadastrados:

ID	Nome
1	Barbeador elétrico Philips bivolt
2	Copo Stanley verde
3	Copo de vinho para vinho tinto
4	Liquidificador elétrico Oster com copo de vidro

A criação das listas começa com a inserção do primeiro produto. Ao criá-lo, as palavras do nome (ou outro atributo) devem ser transformadas em um vetor de palavras. Em seguida, as palavras irrelevantes para as buscas, como os artigos, as preposições e os numerais devem ser descartados desse vetor. Essas palavras irrelevantes são chamadas de *stop words* (palavras vazias). Vocês podem encontrar listas de *stop words* em português na Internet. Em seguida, é necessário aplicar uma transformação para que as palavras restantes fiquem em letras minúsculas e sem acentos. Por exemplo, para o primeiro livro, o seguinte vetor será gerado: ["barbeador", "eletrico", "philips", "bivolt"].

Empregando a lógica do valor TFxIDF (*Term Frequency — Inverse Document Frequency*), o próximo passo é determinar a frequência do termo no nome do produto e qual o inverso da frequência do termo entre os produtos. O valor TF indica a taxa de ocorrências de um determinado termo no nome. Por exemplo, para o primeiro produto, o termo "eletrico" aparece uma vez de um total de quatro termos (25% = 0.25). Já para o terceiro produto, o termo "vinho" aparece 2 vezes em quatro palavras (após descartar as *stop words*) e sua frequência é, portanto, 0.5 (25%).

Já o IDF indica o inverso da taxa de ocorrências de um termo entre os nomes (ou entre as entidades). Por exemplo, o termo "copo" aparece em 3 dos 4 produtos. Como buscamos o inverso dessa taxa, o IDF desse termo será 4/3 = 1.333. No entanto, para equilibrarmos os resultados, aplicaremos a função logarítmo a essa valor e somaremos o valor 1 ao resultado, para que todas as respostas sejam acima de 1. Assim, o IDF do termo será efetivamente: log(4/3)+1 = 1.125

Após a inserção de todos os produtos, chegaremos às seguintes listas invertidas:

- barbeador: (1; 0.25)
- bivolt: (1; 0.25)
- copo: (2; 0.333), (3; 0.25), (4; 0.2)
- eletrico: (1; 0.25), (4; 0.2)

- liquidificador: (4; 0.25)
- oster: (4; 0.25)
- philips: (1; 0.25)
- stanley: (2; 0.333)
- tinto: (3; 0.25)
- verde: (2; 0.333)
- vidro: (4; 0.25)
- vinho: (3; 0.5)

Note que as frequências foram registradas como uma taxa das palavras válidas (que não são *stop words*). Nessas listas não aparecem os valores de IDF, mas apenas os valores de TF. Mas sabemos que temos 4 produtos (que devem estar armazenado em algum outro arquivo) e sabemos o tamanho de cada lista. Assim, podemos calcular o IDF a qualquer hora.

Para fazermos a consulta em uma lista assim, precisamos pedir ao usuário que digite os termos de busca. Por exemplo, suponha que ele digita: "copo de vinho". Quando fizermos o mesmo tratamento que fizemos com os nomes dos produtos, chegaremos ao vetor ["copo", "vinho"]. Em seguida, recuperamos a lista de cada termo e multiplicamos os valores de TF pelos valores de IDF dos termos. O IDF do termo "copo" é 1.125, calculado por meio da função log(4/3)+1 e do termo "vinho" é 1.062. Assim, os valores recuperados já multiplicados pelos IDFs serão:

- copo: [2; 0.375), (3; 281), (4; 0.225)
- vinho: [(3; 0.801)]

Finalmente, geramos uma única lista com os valores que tenham o mesmo ID somados:
[(2; 0.375), (3; 1.082), (4; 0.225)]

Ao ordenarmos essa lista pelo valor de cada ID, teremos a seguinte lista: [3, 2, 4]. O produto de ID 1 não aparece na lista. Assim, os produtos serão apresentados nessa ordem para os usuários. Em um sistema real, geralmente as respostas são apresentadas de 10 em 10 (como no Google).

Neste projeto, vocês devem implementar o índice invertido e a busca exatamente como descrito acima. Lembre-se de que o índice invertido usará as listas invertidas cujo código está disponível. Nesse exemplo, elas devem ser atualizadas sempre que um produto for incluído, excluído ou tiver seu nome alterado.

O QUE DEVE SER FEITO?

- Implementar o índice invertido para os produtos usando as palavras dos nomes dos produtos.
- Alterar as listagens de produtos para buscas por palavras oferecendo respostas ordenadas pelo valor TFxIDF.

Quanto tudo estiver pronto, as buscas por produtos deverão ser por GTIN-13 e por palavras.

FORMA DE ENTREGA

- **Código**: Vocês devem postar o seu trabalho no GitHub e enviar apenas o URL do seu projeto. Criem um repositório específico para este projeto (ao invés de mandar o repositório pessoal de algum de vocês em que estejam todos os seus códigos). Acrescentem um arquivo readme.md ao

projeto que será o relatório do trabalho de vocês (explicado abaixo).

- **Relatório**: O relatório deve começar com a lista dos participantes do trabalho prático e, em seguida, ter uma descrição completa do que o sistema faz. Capturem algumas telas e citem os nomes das classes que foram criadas. Expliquem todas as operações especiais que foram implementadas. O objetivo é que vocês facilitem ao máximo a minha correção, de tal forma que eu possa entender com facilidade tudo aquilo que fizeram e dar uma nota justa. No relatório, vocês devem, necessariamente, responder ao seguinte checklist (copie as perguntas abaixo para o seu relatório e responda sim/não em frente a elas, justificando a resposta quando necessário):
  - O índice invertido com os termos dos nomes dos produts foi criado usando a classe ListaInvertida?
  - É possível buscar produtos por palavras no menu de manutenção de produtos?
  - É possível buscar produtos por palavras na hora de acrescentá-los às listas dos usuários?
  - O trabalho compila corretamente?
  - O trabalho está completo e funcionando sem erros de execução?
  - O trabalho é original e não a cópia de um trabalho de outro grupo?
- **Vídeo de demonstração**: Grave um vídeo de **até 3 minutos** (captura de tela com narração em áudio) mostrando as principais operações do seu sistema. Se o vídeo ficar grande demais para o GitHub, vocês podem publicá-lo no YouTube e compartilhar o link ou usar a própria ferramenta do Canvas para captura de vídeo.

Lembre-se de que, para essa atividade, eu avaliarei tanto o esforço quanto o resultado. Portanto, escrevam o relatório e gravem o vídeo de forma que me ajude a observar o resultado.

Atenção: As respostas incorretas ao *checklist* prejudicaram consideravelmente a nota do grupo. Se vocês disserem que fizeram algo que não foi implementado, a nota final será reduzida em 50% por resposta incorreta (duas respostas incorretas significam a nota zero). Além disso, se vocês disserem que algo está funcionando corretamente, mas a operação não funcionar direito, a nota final será reduzida em 25% por resposta incorreta. Dessa forma, quando necessário, justifiquem as respostas ao *checklist*. A falta do relatório no repositório implicará em perda de 50% dos pontos obtidos na atividade. A falta do vídeo implicará, da mesma forma, em perda de 50% dos pontos. Se os dois faltarem, a nota será, automaticamente, zero.

AVALIAÇÃO

Essa atividade vale 5 pontos. A avaliação será feita por meio do relatório. Dessa forma, um relatório incompleto ou ausente impactará na perda significativa de pontos na avaliação do projeto.

Atenção: Trabalhos copiados de colegas, que não evidenciem um esforço mínimo do próprio aluno, serão anulados.

Se tiver dúvidas sobre o trabalho a fazer, me avise. Não deixe de observar que o URL com o código no GitHub deve ser entregue até o dia especificado na atividade.