Universidade Federal de Uberlândia - UFU
Bacharelado em Sistemas de Informação - Campus Monte Carmelo
GSI521 – Organização e Recuperação da Informação – 2019/2
JOÃO VITOR PEREIRA - 31711BSI024
CESAR HENRIQUE CARDOSO - 31711BSI006
RAFAEL - 31611BSI009

1 Introdução

O trabalho tem como objetivo, efetuar uma busca vetorial a partir da leitura de uma cadeia de caracteres, extraindo os termos da cadeia e aplicando a stemização que foi desenvolvida nas duas primeiras fases do projeto. Desta forma, o objetivo é modularizar um sistema que é capaz de efetuar:

- A retirada de stopwords.
- A retirada de gênero.
- O Calculo de IDF/TF
- A busca vetorial com vetores normalizados
- O produto cartesiano para cada vetor de busca
- O ranqueamento dos termos pesquisados

2 Funcionamento

O sistema vai receber uma lista.txt contendo as palavras de vários documentos, no qual dentro dessa lista pode haver stopwords que devem ser retiradas pelo sistema. Além das stopwords o genêro das palavras também é removido, com o intuito de otimizar as pesquisas futuras. O sistema recebe os termos que o usuário deseja encontrar e calcula o IDF/TF de cada termo que foi encontrado na coleção de documentos. É realizada uma busca vetorial com os vetores normalizados, aplicando-se o produto cartesiano para cada vetor de busca e é retornado ao usuário o ranquamento dos documentos de acordo com os termos pesquisados.

3 Biblioteca usadas

As bibliotecas usadas para utilização de arrays foi a Numpy, que é um pacote para o python que trabalha com arrays e matrizes multidimensionais, esta é bem completa pois possui uma grande quantidade de funções matemáticas para efetuar operações dentro do array, além da biblioteca Pandas , que é um pacote para o python que trabalha com DataFrames. Pypdf para tratamento de documentos pdf e a biblioteca glob. A especificação de cada função utilizada segue abaixo:

- numpy.unique(array): Esta função retorna uma lista cujo os elementos do "array"não possuem mais de uma ocorrência.
- numpy.genfromtxt(diretorio): Esta função lê um arquivo no diretório do parâmetro e retorna uma lista multidimensional(como uma matriz).
- numpy.intersect1d(arrya1,array2): Esta função retorna uma lista contendo a interseção de "array1"e "array2".
- numpy.setdiff1d(arrya1,array2): Esta função retorna uma lista contendo os elementos que estão em "array1"e não estão em "array2".
- pd.readCsv(diretorio, sep=, header=None): Esta função retorna um dataframe contendo todos os dados do arquivo csv lido.
- pd.columns: Esta função tem como objetivo tanto retornar as colunas do dataframe, quanto também para manipulá-las.

- glob.glob(): retorna todos os documentos de um diretório especificado.
- PyPDF2.PdfFileReader: Esta função lê um arquivo pdf específicado para que possa ser posteriormente manipulado.

4 Funções criadas

- verificaStopword(palavr): A função "verificaStorword"tem como objetivo receber uma palavra e realizar uma interação com o usuário para questionar se é stopword ou não (Usuário deve responder com s/n). Existem duas possibilidades:

  1. Usuário digita "s": A função retorna true.
  2. Usuário digita "n": A função retorna false.

- escreverEmArquivo(diretorio,frase): A função "escreverEmArquivo"recebe um diretório e o que deseja escrever. Ela abre o arquivo no diretório(no modo "a+") parâmetro e adiciona a "frase"passada também no parâmetro.
- lerTXT(nomeArquivo): A função "lerTXT"recebe um nomeDeArquivo. Ela busca no diretório do arquivo e retornar este arquivo lido na função.
- zerarArquivo(nomeArquivo): A função "lerTXT"recebe um nomeDeArquivo. Ela busca no diretório do arquivo e retorna este arquivo zerado na função.
- GerarIndiceInvertido(dir): A função "GerarIndiceInvertido"recebe um diretório de arquivo. Ela basicamente pega um dataframe e transforma ele em dicionário, onde a chave principal é o termo que é composto por mais duas outras chaves, sendo a frequência e a lista de documentos.
- mensagemSucesso(): A função "mensagemSucesso"retorna uma mensagem de sucesso para cada processo realizado no sistema.
- retirarStopWords(inn, stopwords): A função "retirarStopWords"basicamente recebe duas listas, uma contendo as palavras e a outra a lista de stopwords. Ela vai retornar as palavras que não estão na interselçao entre essas duas listas.
- alterarGenero(notStops, x): A função "removerGenero"basicamente recebe duas listas, uma contendo as palavras sem stopwords e a outra contendo a lista inteira. Ela vai retornar uma lista contendo as palavras gênero.
- menu(): Interface exibida ao usuário.
- calcTF(): A função "calcTF"calcula o TF do posting e retorna tal valor
- calcIDF(): A função "calcIDF"calcula o IDF dos termos no documento e retorna tal valor
- gerarIDFTFdeDicionarioInvertido(): Essa função retorna uma matriz n (número de termos) por m (1 + número de documentos, o "+1"é porque o vetor de busca do usuário fica na coluna zero) contendo o cálculo de IDF,tf para cada posting. As linhas representam o vetor de idf,tf dos termos e as colunas o vetor dos documentos.
- pesquisarIdfTftermo(): A função "pesquisarIdfTftermo()"retorna o vetor de IDF/TF do termo pesquisado a partir da matriz de IDF,TF.
- pesquisarIdfTfDoc(): A função "pesquisarIdfTfDo"retorna o vetor de IDF/TF de um documento a partir da matriz de IDF,TF.
- buscaVetorial(): A função "buscaVetorial"recebe os termos pesquisados pelo usuário e verifica se estão na coleção. Caso estejam, chama a função de ranquear.
- ranquear(): A função "ranquear"recebe o IDF/TF dos termos encontrados na busca, faz o produto cartesiano do vetor de busca do o vetor dos documentos, retornando o ranking dos documentos. Documentos não especificados no retorno significa que os termos não foram encontrados em tal documento.

- montarVetorbusca(): A função "montarVetorbusca"retorna o vetor de busca de acordo com o IDF do termos pesquisados.
- montarVetoresDistancia(): A função "montarVetoresDistancia"retorna o vetor de distância normalizado para o calculo do produto cartesiano.
- gerarDictdocumentosPdf(): A função "gerarDictdocumentosPdf"pega todos os documentos com a extensão pdf no diretório passado por parâmetro. Lê cada documento e retorna um dicionário na qual a chave é o nome do documento e o valor é o conteúdo do documento em formato string. (nesta função que estamos utilizando as libs pypfd2 e glob)
- buscarTrechoDeTermoNoDoc(): Esta função tem como objetivo pesquisar um trecho onde o termo passado por parâmetro ocorre nos documentos q ela pertence. Ela tenta buscar uma quantidade pequena de palavras antes e após a ocorrência do termo, pesquisando no dicionário de documentos criado a partir da função "gerarDictdocumentosPdf". Obs: nas situações em que o termo no dicionário de índice invertido (o criado a partir da função "gerarIndiceInvertido()") é diferente do termo no documento original, a função não conseguirá encontrar o termo, isso é uma consequência da stemização para gerar o índice invertido e a falta de stemização na pesquisa feita pelo usuário. Nesse tipo de situação, aparecerá a mensagem "Problema na stemização do termo".

5 Conclusão

Nesta fase do trabalho implementamos a busca de termos de forma vetorial, que retornar ao usuário o ranking dos termos encontrados na coleção, além de exibir um trecho do documento do termo encontrado. Foram adicionadas 9 funções para realizar todas as tarefas necessárias. Ademais, o trabalho nos ajudou a compreender como funciona um dos mecanismos de buscas mais utilizados no mercado.