Instruções para executar os algoritmos

Todos os ficheiros a serem codificados devem ser previamente convertidos para ficheiros de texto(.txt)

BaseLine

Executar o ficheiro python chamado “baseLine” irá imprimir os resultados das baseLines dos 4 documentos.

Este ficheiro utiliza algumas das funções desenvolvidas no trabalho anterior(“histograma(P,A)” e “limite_minimo(P,A)”) que têm como objetivo calcular a entropia de cada caracter que depois é multiplicada pelo número de caracteres e dividida por 8 para obter o valor de baseLine em Bytes**.**

Huffman(Algoritmo escolhido para codificar o Random)

Executar o ficheiro python chamado “huff”. Irá codificar o ficheiro que estiver em nomefich (ex “random”) e cria dois ficheiros com o nome (nomefich+"huffmandicio.txt" contém a árvore de huffman, nomefich+"codificadoHuffman.bin" contém a informação codificada).

Este ficheiro começa por gerar a árvore de Huffman e depois codifica a informação do texto utilizando esta árvore no final converte em binário.

O seu decoder está no ficheiro python chamado “huffDecoder”. Este ficheiro lê os ficheiros codificados e reconstrói o documento original e coloca-o em nomefich+"descodificadoHuffman.txt".

Ter em atenção o nome do ficheiro deve ser igual tanto no huff como no huffDecoder (exemplo decoder de random, colocar em nomefich=”random”.

TBW->RLE->Huffman

Executar o ficheiro python chamado “principal”. Irá codificar o ficheiro colocado na primeira linha de código do main utilizando uma TBW e um RLE e irá armazenar esta codificação num ficheiro chamado bwtComRle.txt, para a posterior codificação huffman deverá ser executado o ficheiro python chamado “huff” no qual o nome de ficheiro deve ser alterado para bwtComRle.

TBW->LZW->huffman

Executar o ficheiro python chamado “TBW+LZW+HUFF”. Irá codificar o ficheiro que estiver na primeira linha do código do main. Escreve a codificação em "LZW+MOVE+HUFF.txt".

Começa por aplicar um algoritmo de TBW, posteriormente aplica um LZW e ao resultante é aplicado codificação huffman.

LZW-> Myhuffman

Executar o ficheiro python chamado “LZW+myhuffman” irá codificar o ficheiro que estiver na variável nome (primeira linha do main) e coloca o ficheiro codificado em nome+"LZW+myhuff.bin".

Começa por codificar a informação usando um algoritmo de LZW e posteriormente utiliza o algoritmo Myhuffma(codificação utilizando uma árvore binária fixa).

LZW->huffman (utilizado no jquery)

Executar o ficheiro python chamado “lzw+huffman” ” irá codificar o ficheiro que estiver na variável nome(primeira linha do main e coloca o ficheiro codificado em em nome+"dicionario.txt".

Começa por codificar a informação usando um algoritmo de LZW e posteriormente utiliza o algoritmo de huffman.

O seu decoder está no ficheiro python chamado “decoder LZW+huffman”. Este ficheiro lê os ficheiros codificados e reconstrói o documento original e coloca-o em nome+"descomprimido.txt". A variável nome deve ser alterada para o nome do ficheiro que se pretende descodificar(ex “jquery-3.6.0).

VarianteCompress->myhuffman

Executar o ficheiro python chamado “Compressor” que irá codificar o ficheiro que estiver na variável nome. Coloca o ficheiro codificado em 3 ficheiros (nomeFicheiro+"Comprimido.bin" (tem a informação), "dicionarioMetodo.txt" (tem a árvore de huffman do dicionário), "DicionarioInformacao.bin"(tem o dicionário codificado).

Ter em atenção para o finance é usado uma vírgula como separador, ou seja para alternar entre codificar o finance e o bible deve se abrir o ficheiro python “métodoCompressWordIndex” e na função variantecompress nas duas primeiras linhas descomentar a linha correspondente e comentar a outra.

Apesar de não ser utilizado para codificar nenhum dos ficheiros criamos um decoder para este algoritmo. Executar o ficheiro python chamado “Decoder”. O nome do ficheiro tem de ser o nome do ficheiro que queremos descodificar e o ficheiro descodificado é guardado em nomeFicheiro+"descomprimido.txt.

Tendo em atenção que o caractere usado para separador deve também ser alterado no ficheiro “decompresse meudicio+huffman” no último ciclo for do main (linhas68-69).

VarianteCompress->huffman(utilizado para bible e finance)

Executar o ficheiro python chamado meudicio+huffman que irá comprimir o ficheiro que estiver em nomefich. O ficheiro codificado é colocado em 4 ficheiros (nomefich+"dicionario.txt"(arvore de huffman da informação codificada), nomefich+"informacao.bin"(informação codificada), "dicionarioMetodo.txt" (tem a árvore de huffman do dicionário), "DicionarioInformacao.bin"(tem o dicionário codificado)).

Ter em atenção para o finance é usado uma vírgula como separador, ou seja para alternar entre codificar o finance e o bible deve se abrir o ficheiro python “métodoCompressWordIndex” e na função variantecompress nas duas primeiras linhas descomentar a linha correspondente e comentar a outra.

Para decodificar, executar o ficheiro python chamado decompresse meudicio+huff, o nome do ficheiro tem de ser o nome do ficheiro que queremos descodificar.

Irá ler os ficheiros resultantes da codificação e guardar o texto decodificado em nomeFicheiro+"descomprimido.txt".

Tendo em atenção que o caractere usado para separador deve também ser alterado no ficheiro “decompresse meudicio+huffman” no último ciclo for do main (linhas68-69).