

Teoria da Informação

Estados de Artes Codec de texto

- Bruno Sequeira nº2020235721
- Rui Santos nº2020225542
- Rodrigo Figueiredo nº2020236687

Taxa de Compressão

Ao longo deste projeto, testamos vários algoritmos de compressão.

Contudo, calculamos a taxa de compressão dos algoritmos pela seguinte formula:

$$\text{Compression Ratio} = \frac{\text{Uncompressed Data Rate}}{\text{Compressed Data Rate}}$$

Fig. - Formula de rácio de Compressão

Obs.:

Uncompressed Data Rate - Tamanho do ficheiro original.

Compressed Data Rate - Tamanho do ficheiro comprimido.

Representação do Rácio de Compressão

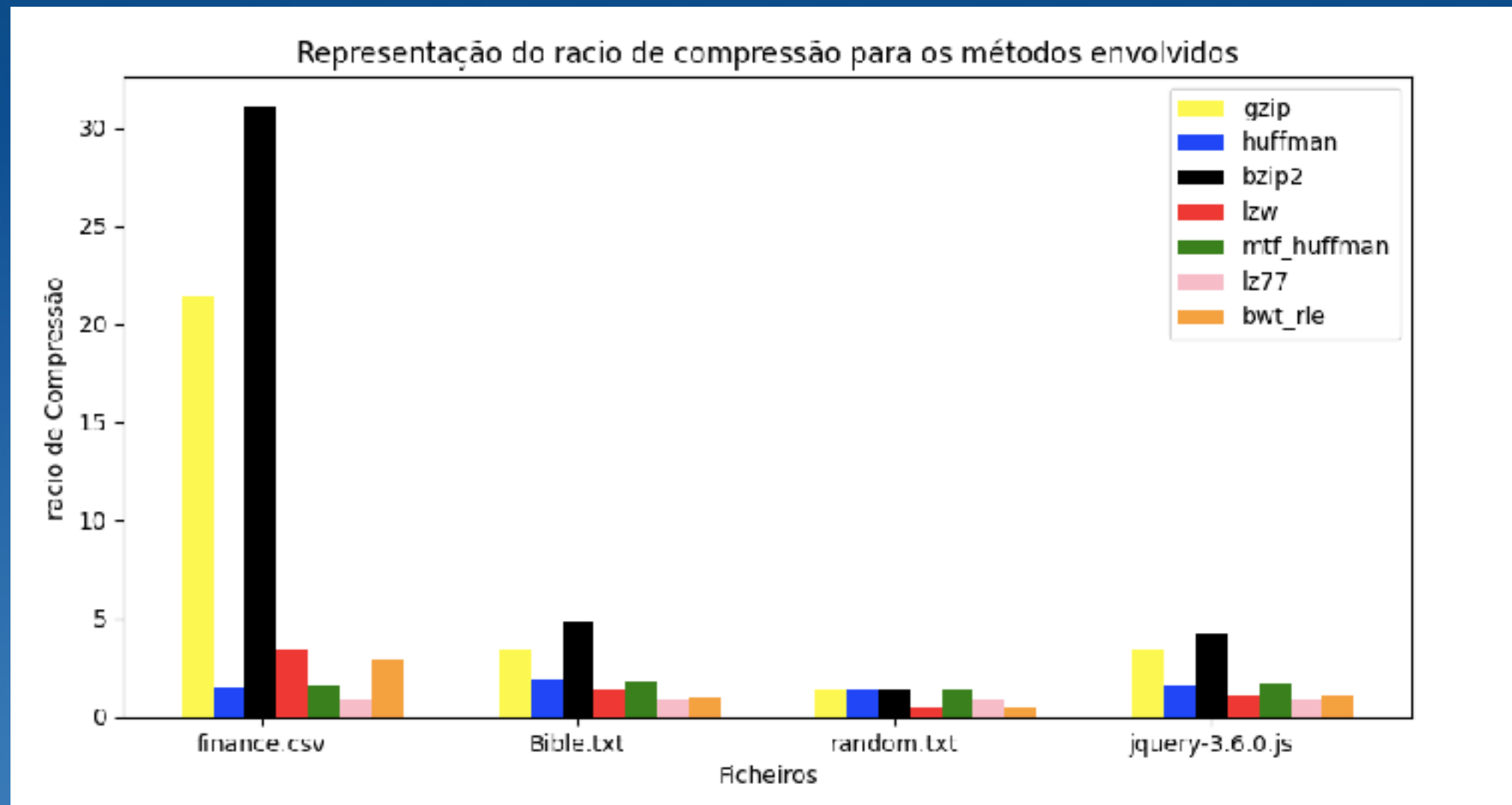


Fig. Representação do rácio de compressão para os métodos envolvidos.

Valores do Rácio de Compressão

Ficheiros/ métodos	Huff.	Mtf+huff	Bzip2	gzip	lzw	Lz77	Bwt+rle
Finance.csv	1.55	1.595	31.045	21.393	3.50	0.925	2.997
Bible.txt	1.82	1.789	4.786	3.431	1.36	0.947	1.023
Random.txt	1.33	1.351	1.321	1.319	0.52	0.889	0.524
Jquery- 3.6.0.js	1.56	1.653	4.190	3.376	1.11	0.965	1.110

Fig. - Tabela com Valores do Rácio de Compressão.

Análise de resultados:

Finance.csv - Bzip2, com rácio (31.045).

Bible.txt - Bzip2, com rácio (4.786).

Random.txt - 4 métodos com rácios parecidos.

Jquery-3.6.0.js - Bzip2, com rácio (4.190).

Representação do tempo de Compressão

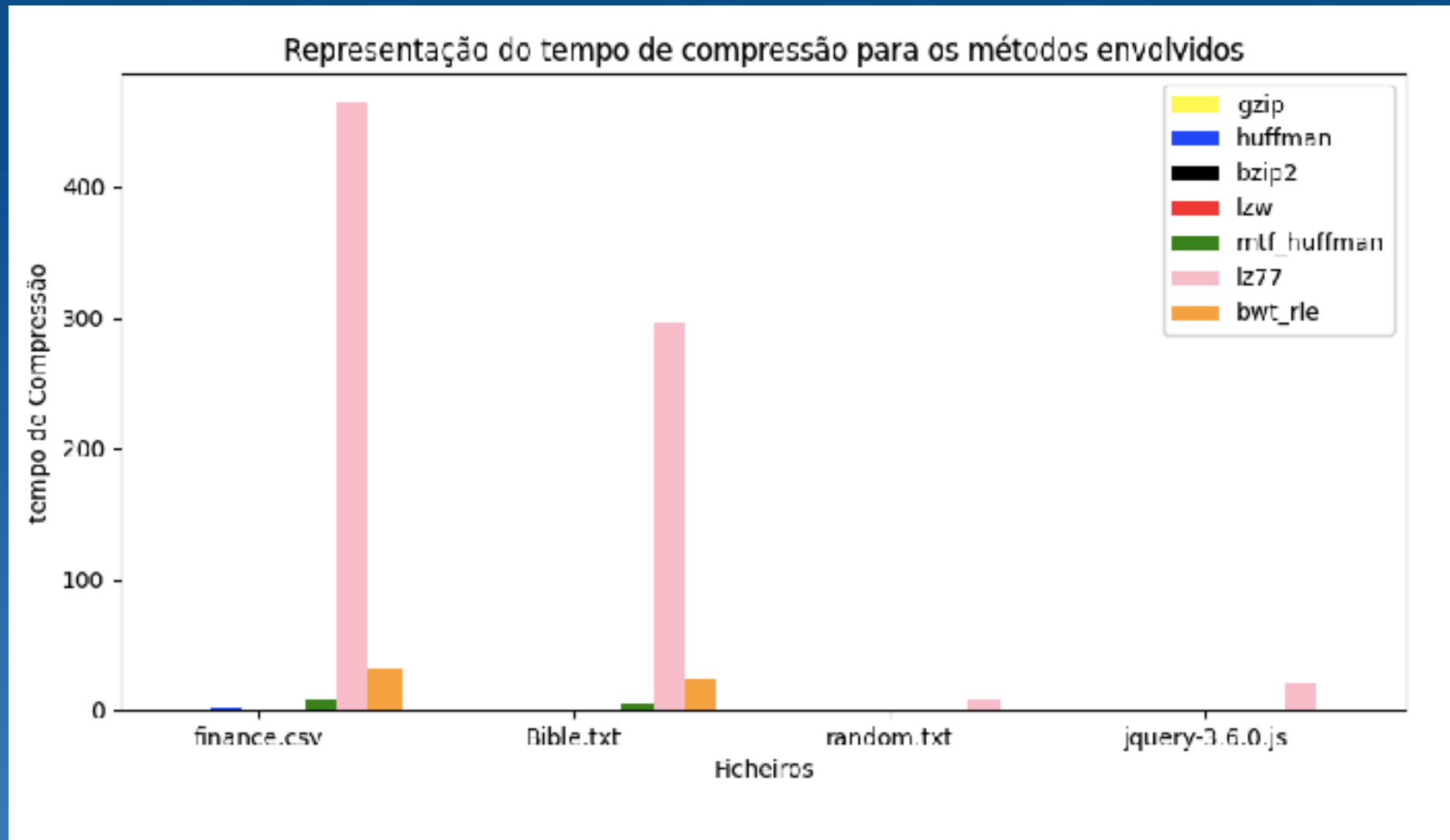


Fig. - Representação do tempo de Compressão.

Valores do Tempo de Compressão

Ficheiros/ métodos	Huff.	Mtf+huff	Bzip2	gzip	lzw	Lz77	Bwt+rle
Finance.csv	2.26	8.70	5.53	0.09	1.55	463.8	32.63
Bible.txt	1.40	4.82	0.30	0.45	1.05	296.6	24.23
Random.txt	0.04	0.15	0.01	0.005	0.03	8.42	0.63
Jquery- 3.6.0.js	0.10	0.36	0.02	0.03	0.06	20.60	1.69

Fig. Tabela com os Valores do Tempo de Compressão em Segundos.

Para um método de codificação ser eficiente e relativamente bom, o tempo de compressão tem de ser o mínimo possível.

Análise da tabela :

Finance.csv- gzip com tempo de 0.09s.

Bible.txt - Bzip2 com tempo de 0.30s.

Random.txt - Bzip2 com tempo de 0.01s.

Jquery-3.6.0.js - Bzip2 com tempo de 0.02s.

Melhores métodos de Compressão

Analisemos agora os melhores métodos para os ficheiros estudados:

- **Finance.csv** - O melhor método de compressão é o Bzip2. Tendo o maior rácio de compressão, apesar do tempo não ter sido o melhor, comparando com os demais métodos continua a ter uma boa eficiência.
- **Bible.txt** - O melhor método foi também o Bzip2 assim como o seu tempo de compressão.
- **Random.txt** - Pela análise dos resultados da tabela (rácio), 4 dos métodos aplicados obtiveram rácios de compressão bastante próximos, logo analisamos os tempos. Concluímos assim que o gzip teve um tempo de compressão bastante curto, sendo este um dos melhores métodos aplicados neste ficheiro.
- **Jquery-3.6.0.js** - O melhor método foi o Bzip2.

Conclusão

- Concluimos então que o melhor método de compressão, entre todos os referidos neste artigo, é o Bzip2, tendo sido o melhor, em todos os casos, em questão de rácio. O que faz sentido, sendo um dos métodos de compressão mais usados para compressão de dados. Sendo utilizado pelo 7zip, WinRar, etc.

- Trabalho futuro sugerido:

Encontrar um método de compressão ainda mais eficiente, que sirva universalmente para todos os tipos de dados.

Bibliografia

- Gzip - <https://docs.python.org/3/library/gzip.html>
- RLE - <https://stackabuse.com/run-length-encoding/>
- Huffman - <https://towardsdatascience.com/huffman-encoding-python-implementation-8448c3654328>
- BWT- <https://github.com/TiongSun/DataCompression/blob/master/BWT.ipynb>
- LZW - <https://stackoverflow.com/questions/6834388/basic-lzw-compression-help-in-python>
- Move to front - https://en.wikipedia.org/wiki/Move-to-front_transform
- Bzip2 - <https://docs.python.org/3/library/bz2.html>
- LZ77 - <https://github.com/manassra/LZ77-Compressor>