Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

DCC012
ESTRUTURAS DE DADOS II

Trabalho 3 - Compressão de Tweet

Felipe Barra Knop - 201565553C

Lohan Rodrigues N. Ferreira - 201565082AC

Lucas Carvalho Ribeiro - 201565554AC

Professora Vânia de Oliveira Neves

Juiz de Fora - MG
4 de dezembro de 2017

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

Neste relatório faremos uma análise sobre quatro tipos de algoritmos de compressão de arquivos diferentes, sendo eles Huffman, LZ77, LZ78 e LZW. Faremos comparações entre os algoritmos de modo a verificar seus desempenhos conforme os seguintes parâmetros: Taxa de Compressão, Tempo de Processamento e Tamanho Médio do Arquivo em Disco.

A Taxa de Compressão foi calculada da seguinte forma:

tamanho do conteúdo original/tamanho do conteúdo comprimido

1.1 Considerações iniciais

- Ambiente de desenvolvimento do código fonte: IntelliJ IDEA Community Edition.
- Linguagem utilizada: Java 8.
- Ambiente de desenvolvimento da documentação: ShareLaTeX - Editor online de LaTeX.

1.2 Dados utilizados

Todos os testes documentados neste relatório foram realizados com um conjunto real de Tweets públicos. A lista utilizada aqui possui 1.000.000 registros de Tweets diferentes e pode ser obtida no seguinte endereço:
`https://www.dropbox.com/s/07zflza2hj6njzj/tweets.txt?dl=0`

1.3 Testes

Todos os testes documentados neste relatório foram realizados da seguinte forma: para cada valor de N (1.000, 5.000, 10.000, 50.000, 100.000, 500.000 e 1.000.000) foram gerados 5 conjuntos de N elementos aleatórios sorteados com 5 sementes diferentes. Os resultados inclusos nas tabelas são formados pela média dos resultados de cada conjunto gerado para aquele número N de elementos.

Os resultados documentados de cada teste são como descritos na seção 1.

1.4 Realização das Atividades

Todos os membros do grupo contribuíram de alguma forma em todas as partes do trabalho, mas a divisão de tarefas foi, principalmente, a seguinte:

- Felipe: Huffman e LZ77
- Lohan: LZW e Relatório
- Lucas: LZ78 e Relatório

2 Análise de Algoritmos de Compressão

Nesta seção analisaremos por partes o desempenho dos 4 algoritmos de compressão anteriormente citados.

Devido ao grande número de dados utilizados nos testes, não foi possível executar todos os algoritmos com as quantidades descritas na seção 1.3, pois estes apresentaram restrições de tempo e/ou memória ao serem executados com quantidades de dados muito grandes, tanto por sua complexidade devido ao casamento de padrões em cadeias de caracteres quanto pelo tamanho dos dicionários guardados em memória.

Os resultados podem ser vistos nas tabelas 1, 2, 3:

	Tempo(ms)			
N	Huffman	LZ77	LZ78	LZW
1000	16	1053	5186	10953
5000	46	4835	97587	161332
10000	22	9656	-	-
50000	259	48873	-	-
100000	327	99248	-	-
500000	1259	-	-	-
1000000	2835	-	-	-

Tabela 1: Tempo médio gasto

	Taxa de Compressão			
N	Huffman	LZ77	LZ78	LZW
1000	1,48	1,00	1,07	1,08
5000	1,48	1,01	1,27	1,35
10000	1,48	1,00	-	-
50000	1,48	1,01	-	-
100000	1,48	1,01	-	-
500000	1,48	-	-	-
1000000	1,48	-	-	-

Tabela 2: Taxa de compressão

	Tamanho do arquivo(bytes)			
N	Huffman	LZ77	LZ78	LZW
1000	87435	128659	120922	119373
5000	437049	643153	507370	447513
10000	874798	1287322	-	-
50000	4372655	6436537	-	-
100000	8747082	12875059	-	-
500000	43752379	-	-	-
1000000	87491266	-	-	-

Tabela 3: Tamanho do Arquivo na Memória

Devido à natureza dos dados (textos com linguagem natural) os algoritmos de compressão baseados em repetição de sequências de caracteres (LZ77, LZ78, LZW) não se mostraram muito eficientes, enquanto que o algoritmo de Huffman se mostrou várias vezes mais rápido e mais eficiente que os outros, o que era esperado já que não depende da repetição de cadeias de caracteres, e sim da frequência de cada caracter separado.

A taxa de compressão do algoritmo Huffman se mantem constante independente da quantidade de tweets, isso ocorre pois em linguagem natural a variação com que os caracteres aparecem em um texto seguem um padrão, que provavelmente não mudará independente de quantos tweets sejam colocados juntos numa compressão e como essa frequência não muda é esperado que independente do tamanho do arquivo, a compressão se mantenha.

Algo importante a ser notado no LZ77 é que sua compressão não foi muito eficiente independente do tamanho dos dados, isso acontece pois, diferentemente das outras versões do LZ, o LZ77 não possui um dicionário capaz de guardar muitas informações e, como a natureza dos dados (forma como tweets são escritos) nos leva a crer que poucas cadeias se repetirão em uma pequena janela, a compressão quase não acontece. Porém o aumento dessa janela, e do buffer do LZ77 são capazes de tornar o código mais eficiente ao custo de mais memória no processamento e consequentemente no tempo, o que acarretaria aos problemas encontrados no processamento dos outros LZs (LZ78 e LZW) que foram incapazes de comprimir arquivos extremamente grandes por exigir alto consumo de memória e tempo.

A taxa de compressão dos algoritmos LZ78 e LZW são muito semelhantes, o que pode ser visto na tabela 2. Isso era esperado visto que a forma como fazem compressão são aproximadamente iguais tendo somente algumas diferenças na forma como constroem seus dicionários.

3 Conclusão

Sem muita análise já fica claro que o algoritmo que melhor se adaptou à compressão do tipo de arquivo usado neste trabalho é o Huffman, que como citado anteriormente não depende das repetições de cadeias e sim de repetições de carac-

teres, se sobressaindo no contexto de comprimir *Tweets*.

Foi possível perceber na implementação também, em relação ao algoritmo LZ77, uma possível definição de parâmetro de qualidade que pode ser feita dinamicamente na hora de começar a compressão, que não está presente nos outros algoritmos. Ao definir um número para o tamanho da janela de procura, é possível escolher se o arquivo deve ser comprimido com mais agilidade, sacrificando um pouco da taxa de compressão, ou comprimido mais devagar, mas com um resultado mais satisfatório em relação à taxa de compressão, com valores menores e maiores para o tamanho da janela, respectivamente.