Tópicos em Recuperação de Informação[1]

Nivio Ziviani

[1]Conjunto de transparências elaborado por Nivio Ziviani e Patrícia Correia

Compressão de Texto

- Compressão de texto significa encontrar formas para representar o texto em poucos *bits* ou *bytes*.
- Ponto importante em um ambiente de RI.
- Escolha atrativa, se não obrigatória.
- Ganho: menos espaço de armazenamento e menos tempo para ser transmitido.
- O preço a pagar: tempo para codificar e decodificar o texto.

Compressão de Texto e Sistemas de RI

- O maior obstáculo para armazenar texto na forma comprimida é a necessidade dos sistemas de RI em acessar o texto randômicamente .

- Necessário decodificar o texto inteiro do início até que a palavra desejada seja alcançada.

- Tópicos em métodos de compressão de texto que são desejáveis para serem usados em uma ambiente de recuperação de informação:
  - acesso randômico barato para palavras dentro do texto comprimido
  - permitir descompressão em pontos intermediários no texto comprimido.
  - buscar diretamente sobre o texto comprimido comprimindo o padrão.

Métodos de Compressão

Características importantes:

- Economia de espaço (taxa de compressão acima de 25%).

- Velocidade de compressão e descompressão. Em algumas situações a velocidade de descompressão é mais importante que a velocidade de compressão.

- Capacidade de casamento de padrão comprimido:
  - definido como a tarefa de realizar casamento de padrão em um texto comprimido sem descomprimi-lo.
  - busca seqüencial pode ser acelerada comprimindo-se a chave de busca ao invés de decodificar o texto comprimido.
  - como uma conseqüência disso, muito menos texto tem que ser examinado.

Fases do Processo de Compressão

Modelagem

A probabilidade é atribuída para cada próximo símbolo.

Codificação

Próximo símbolo é representado pelo seu código relacionado à distribuição de probabilidade obtida pela fase de modelagem.

Definições

1. Um *símbolo* aqui é usualmente um caractere, uma palavra ou um número fixo de caracteres.

2. O conjunto de todos os possíveis símbolos no texto é chamado *alfabeto*.

3. *Taxa de compressão* é o tamanho do arquivo comprimido como uma porcentagem do arquivo descomprimido.

4. Um *método de compressão de ordem zero* opera símbolo por símbolo com nenhum contexto usado para predizer o próximo símbolo, e assim cada símbolo de entrada é tratado como um símbolo independente.

Abordagens para Compressão de Texto

Métodos Estatísticos

Uma probabilidade é obtida para cada símbolo e o código obtido é baseado na probabilidade.

Métodos de Dicionário

Substitui uma seqüência de símbolos por um ponteiro que aponta para uma ocorrência anterior da seqüência.

Métodos Estatísticos

- Contam com a geração de boas estimativas de probabilidade para cada símbolo.

- Quanto mais precisas as estimativas são, melhor é a compressão obtida.

- A tarefa de estimar a probabilidade sobre cada próximo símbolo é chamada de *modelagem*.

- Uma vez que estas probabilidades estão disponíveis, os símbolos são convertidos em dígitos binários, um processo chamado de *codificação*.

Métodos Estatísticos

- Na prática, ambos o codificador e decodificador usam o mesmo modelo.

- O decodificador interpreta a saída do codificador (com referência ao mesmo modelo) para encontrar o símbolo original.

Modelagem e codificação são atividades distintas

Métodos Estatísticos

- Tarefa de modelagem: a probabilidade é estimada para cada símbolo.
- Tarefa de codificação: baseada nesta probabilidade, um código é atribuído para cada símbolo por vez.
- Códigos menores atribuídos para os símbolos mais prováveis.

Uso do Modelo na Prática

Codificador

Gera códigos usando o modelo.

Decodificador

Interpreta os códigos com referência ao modelo.

Codificador e decodificador usam o mesmo modelo

Uso do Modelo na Prática

- Codificação: representação de um símbolo com a probabilidade p em aproximadamente $\log \frac{1}{p}$ bits.

- Codificador: produz um conjunto de bits ou bytes do conjunto de probabilidades que governa a escolha do próximo símbolo.

- Decodificador: decodifica o símbolo usando o mesmo conjunto de probabilidades usado pelo codificador.

Métodos de Dicionário

- Usa representação de ponteiros para codificar referências para entradas no dicionário.
- Dicionário é uma lista de símbolos (geralmente chamado frases) que são esperados ocorrerem freqüentemente.
- Ponteiros para entradas de dicionários necessitam menos espaço que a frase que eles substituem, obtendo assim a compressão.

  Ex.: jan, feb,$\cdots$, dec são codificados como 1, 2, $\cdots$, 12
- A distinção entre modelagem e codificação não existe em métodos de dicionários.
- Nenhuma probabilidade explícita associada as frases.
- Métodos de dicionários mais conhecidos: família Ziv-Lempel.

Métodos de Dicionário

- Alcança compressão trocando grupos de símbolos consecutivos (ou frases) por um ponteiro para uma entrada no dicionário.

- Decisão central: seleção das entradas no dicionário.

- A escolha de frases pode ser feita através de esquemas estáticos, semi-adaptativos ou adaptativos.

- Família Ziv-Lempel de esquema de dicionário:
  - troca strings de caracteres com uma referência para uma ocorrência anterior da string.
  - se um ponteiro para uma ocorrência de uma string é armazenado em menos bits que a string que ele substituiu então a compressão é alcançada.

Métodos de Dicionário

Desvantagens sobre os Métodos Estatísticos

- Não é possível iniciar a decodificação no meio do arquivo comprimido.

- Acesso direto para uma posição no texto comprimido não é possível.

$$\Downarrow$$

Métodos de dicionário não são adequados para sistemas de RI

Relacionamento entre Probabilidades e Códigos

- Claude Shannon (1948) em seu teorema:

  Em um esquema de codificação ótimo, um símbolo que é esperado para ocorrer com probabilidade p deve ser atribuído um código de tamanho $\log_2 \frac{1}{p}$ bits.

- O número de bits no qual o símbolo é melhor codificado representa o conteúdo informacional do símbolo.

- A quantidade média de informação por símbolo sobre o alfabeto todo é chamado de *entropia* da distribuição de probabilidade, que é dada por:

$$E = \sum p_i \log_2 \frac{1}{p_i}$$

  Medida em bits por símbolo e representa um limite que nunca pode ser batido em um dado modelo.

Entropia

- Quantidade de ordem (ou redundância) em uma mensagem.
  - pequena: alta ordem.
  - grande: alta desordem.

- Idealmente, o tamanho da codificação deve ser igual a entropia da mensagem .

Modelos de Compressão

- *Estático*: mesmo modelo para todos os textos de entrada.

- *Semi-estático*: diferentes modelos para cada texto.
  - aprender distribuição de prob. no primeiro passo.
  - comprimir texto no segundo passo usando código fixo derivado da distribuição aprendida no primeiro passo.
  - enviar modelo para o decodificador antes da mensagem codificada.

- *Adaptativo*: inicia com nenhuma informação sobre o texto e aprende sobre sua distribuição de probabilidade.
  - uma passagem sobre o texto.
  - não necessita enviar o modelo em separado para o decodificador.
  - código usado para um símbolo particular é baseado no texto já codificado.
  - maior desvantagem: descompressão tem que iniciar do começo e nenhum acesso randômico é possível.

Modelo Baseado em Palavra

- Considera palavras ao invés de caracteres como símbolos. Strings de caracteres no conjunto {A..Z, a..z} separadas por caracteres que não estão no conjunto {A..Z, a..z}.

- Melhores taxas de compressão são alcançadas (palavras carregam muito do significado em linguagem natural).

- Palavras são os átomos sobre os quais a maioria dos sistemas de recuperação de informação são construídos.

- Freqüências de palavras também são úteis para responder consultas envolvendo combinação de palavras.

Escolha do Método de Codificação

- Método de codificação mais conhecido: Huffman (1952).
  - usualmente usado com modelagem estática ou semi-estática.
  - pode ser usado com modelagem adaptativa.

Escolha do Método de Codificação

Codificação Aritmética:

- Computa o código incrementalmente, um símbolo por vez.
- Natureza incremental não permite iniciar a decodificação no meio do arquivo comprimido.
- Esta característica faz este método inadequado para ser usado em um ambiente de recuperação de informação.

Algoritmo da Árvore de Huffman

1. Liste todos os possíveis símbolos com suas probabilidades.

2. Considere os dois símbolos com as menores probabilidades.

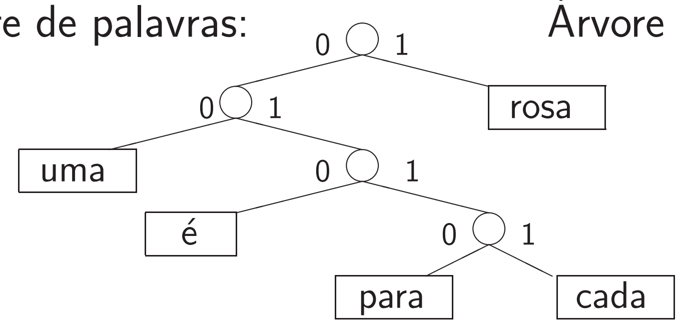
3. Substitua estes dois por um único conjunto, cuja probabilidade é a soma das probabilidades individuais.

4. Repita até que a lista contenha somente um membro.

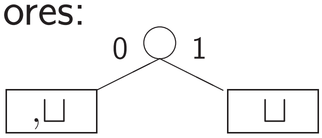
Esquema de Compressão

Alfabetos Separados:

Árvore de palavras:



Árvore de separadores:



Texto original: para cada rosa, uma rosa é uma rosa

Texto comprimido: 0110 1 0111 1 1 0 00 1 1 1 010 1 00 1 1

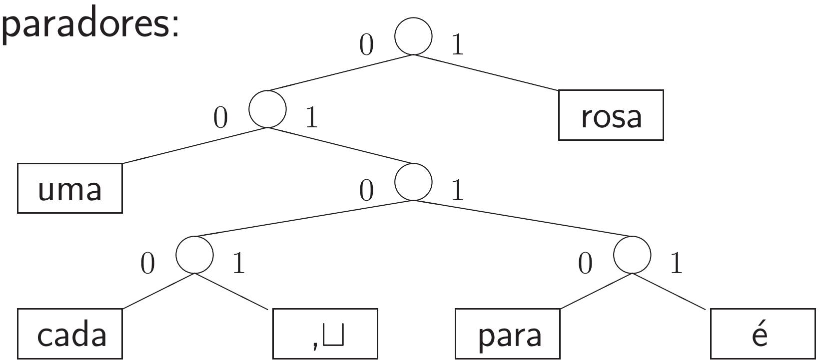
- Texto pode ser reconstruído de qualquer posição.

- Taxa de compressão: 25% a 30%.

Esquema de Compressão

Alfabeto Único:

Árvore de palavras + separadores:



Texto original: para cada rosa, uma rosa é uma rosa

Texto comprimido: 0110 0100 1 0101 00 1 0111 00 1

Arquivos	Palavras (p)	Espaços simples(s)	s/p
AP	38.977.670	30.020.803	0,77
DOE	28.505.125	22.001.366	0,77
FR	34.455.982	25.506.763	0,74
WSJ	42.710.250	33.909.610	0,79
ZIFF	39.675.248	27.225.864	0,69

Árvore de Huffman Canônica

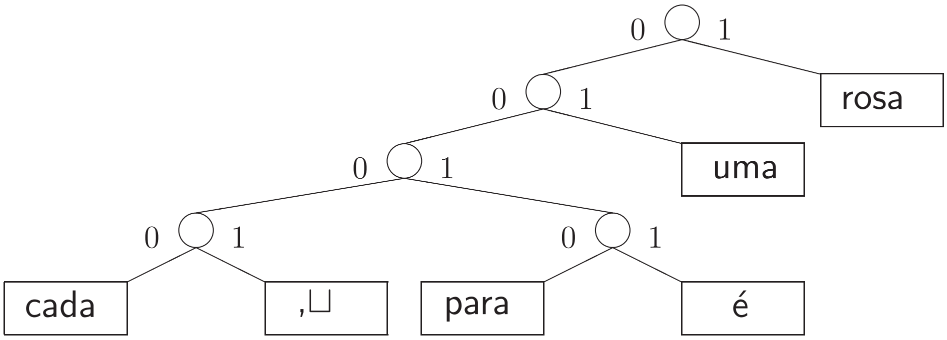
- O número de árvores de Huffman para uma dada distribuição de probabilidade é grande.

- Trocando sub-árvores da esquerda e da direita de qualquer nodo interno resulta em uma árvore diferente na estrutura, mas tamanho do código médio não é afetado.

- Ao invés de usar qualquer tipo de árvore é adotado uma *árvore canônica*, a qual impõe uma ordem particular para a codificação de bits.

- Uma árvore de Huffman é canônica quando a altura da sub-árvore esquerda de qualquer nodo nunca é menor que a sub-árvore direita, e todas as folhas estão em ordem crescente de probabilidades da esquerda para a direita.

Árvore de Huffman Canônica



Símbolo	Prob.	Código Antigo	Código Can.
cada	$1/9$	0100	0000
,␣	$1/9$	0101	0001
para	$1/9$	0110	0010
é	$1/9$	0111	0011
uma	$2/9$	00	01
rosa	$3/9$	1	1

Árvore de Huffman Canônica

- Folha mais profunda na posição mais a esquerda (elemento com menor probabilidade) conterá somente zeros.
- Os códigos seguintes estarão em ordem crescente dentro de cada nível.
- A cada mudança de nível desloca-se um bit para a esquerda no contador.

Código de Huffman Canônico

- Seqüência ordenada S de pares (x_i, y_i), $1 \leq i \leq \ell$, onde x_i representa o número de elementos no nível i e y_i representa o valor numérico do primeiro código no nível i.

- Para o exemplo, a seqüência ordenada é:
$S = \langle(1, 1), (1, 1), (0, \infty), (4, 0)\rangle$

- O par $(4, 0)$ em S corresponde ao quarto nível e indica que existem quatro nodos neste nível e que o nodo mais a esquerda é atribuído código com valor 0 (código 0000 para o quarto nível).

- Conjuntos de códigos tendo o mesmo tamanho são representações binárias de inteiros consecutivos.

- Interpretados como inteiros, os códigos de 4 bits são 0, 1, 2 e 3, código de 2 bits é 1 e código de 1 bit também é 1.

Código de Huffman Canônico

Como decodificar:

- Se o primeiro caractere é 1, um código foi identificado e o símbolo correspondente pode ser decodificado.

- Se este valor é 0, um segundo bit é adicionado e os dois bits são novamente interpretados como um inteiro e usado para indexar a tabela e identificar o símbolo correspondente.

- Uma vez que nós lemos "00" nós sabemos que o código tem quatro bits e portanto nós podemos ler mais dois e usá-los como um índice dentro da tabela.

- Este fato pode ser explorado para possibilitar eficiente codificação e decodificação com pequeno *overhead*. Além disso, muito menos memória é requerida, o que é especialmente importante para grandes vocabulários.

Referências

- Ziviani, N. and Ribeiro-Neto, B. "Text Operations". In: Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval*, Chapter 7, Addison-Wesley, 1999, 28 pages.

- Moura, E. S., Navarro, G., Ziviani, N. and Baeza-Yates, R. "Fast Searching on Compressed Text Allowing Errors", *21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, August 1998.