Compresión

UCR – ECCI

CI-2414 Recuperación de Información

Prof. Kryscia Daviana Ramírez Benavides

Introducción

- Grandes cantidades de información textual en la Internet.
- Se desea representar esta información con menos bits o bytes.
- Esto se logra con una representación reducida del texto utilizando estructuras que existen en éste y algoritmos que las aprovechan.
- La compresión de palabras y no la de caracteres (compresores de uso general).

Técnicas Básicas

- Algoritmos simples de compresión:
  - Estadísticos:
    - Código de Huffman.
    - Código Aritmético.
  - Basados en diccionario:
    - Familia de Ziv-Lempel.
- En la siguiente presentación se verá el algoritmo de compresión estadístico *Código de Huffman*.

Métodos Estadísticos

- Generan buenas estimaciones de la probabilidad que tiene cada símbolo de aparecer en el texto.
- El modelaje es estimar la probabilidad de aparición de cada palabra.
- La codificación óptima para un símbolo con probabilidad p, se espera que su código sea de $log_2 1/p$ bits.
- El número de bits en los que cada símbolo se codifica representa la información contenida en el símbolo.
- La entropía de la distribución de la probabilidad es la cantidad promedio de información por símbolo.

Métodos Estadísticos (cont.)

- La entropía define la longitud media de código mínima necesaria para poder transmitir un mensaje sin perder información (límite inferior en la compresión, medido en bits por símbolo):

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

- La expresión matemática se puede traducir como:
  - Por cada uno de las palabras diferentes que aparece en el texto, se tiene que hallar el logaritmo en base al orden del código de la inversa del valor de la probabilidad de que aparezca en el texto ése carácter.
  - El valor obtenido del logaritmo, se multiplica por la probabilidad de aparición de esa palabra.
  - Estos dos pasos se tienen que hacer por cada una de las palabras diferentes que aparecen en el texto y al final se suma todos los valores obtenidos.

Métodos Estadísticos – Modelaje

- Adaptativo:
  - Comienzan sin información acerca del texto y progresivamente aprenden acerca de su distribución estadística.
  - Da una sola pasada al texto.
  - Si existen grandes cantidades de documentos se aproxima a distribuciones reales.
  - No permite acceso aleatorio del texto.
- Estático:
  - Se asume una distribución, que se calcula una vez, para todos los textos.
  - Si las estructuras del texto difieren mucho de la distribución calculada al principio, se obtienen malos resultados.
- Semi-Estático
  - No asume una distribución inicial.
  - En la primera pasada se “aprende” la distribución.
  - En la segunda pasada se comprime la información, utilizando un código mejorado derivado de la distribución aprendida en la primera pasada.

Métodos Estadísticos – Codificación

- La tarea de obtener la representación de un símbolo basado en una probabilidad de distribución dada por un modelo.
- La meta principal: para asignar códigos cortos a los símbolos más probables (que más aparecen) y códigos largos a los más improbables (que menos aparecen).
- A veces, es necesario sacrificar la proporción de compresión para reducir el tiempo de codificar y decodificar el texto.
- Existen dos estrategias de codificación:
  - Código de Huffman.
  - Código Aritmético.

Código de Huffman

- Código óptimo dentro de los códigos de codificación estadística, es el código de menor longitud media.
- A los símbolos con mayor frecuencia de aparición se les asignarán las palabras de código binario de menor longitud.
- Se ordena el conjunto de símbolos del alfabeto fuente en orden creciente de probabilidades de aparición.
- Se juntan los dos símbolos con menor probabilidad de aparición en un único símbolo, cuya probabilidad será la suma de las probabilidades de los símbolos que lo originaron.
- Se repite este proceso hasta que sólo tengamos dos símbolos.
- Se asigna un 1 a uno de los dos símbolos que tenemos y un 0 al otro.

Código de Huffman (cont.)

- Recorreremos la estructura que hemos construido hacia atrás, cuando dos símbolos hayan dado origen a un nuevo símbolo, estos "heredarán" la codificación asignada a este nuevo símbolo.
- Se le añadirá un 1 a la codificación de uno de los símbolos y un 0 a la del otro símbolo.
- Sustituimos cada palabra del texto por el código respectivo y, una vez hecho esto, agrupamos los bits en grupos de ocho, es decir en bytes.

Ejemplo del Código de Huffman

- Se obtienen las frecuencias de cada palabra dentro del documento:

casa	29
nuevo	7
pesa	12
plato	5
sucio	4
tarde	8

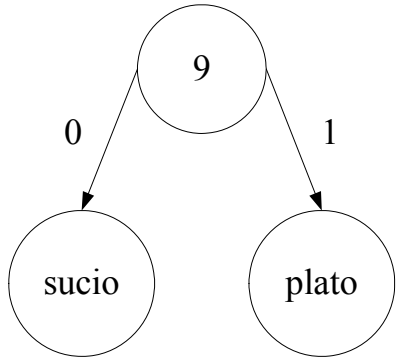
- Se ordenan las frecuencias en orden ascendente:

(sucio, plato, nuevo, tarde, pesa, casa)

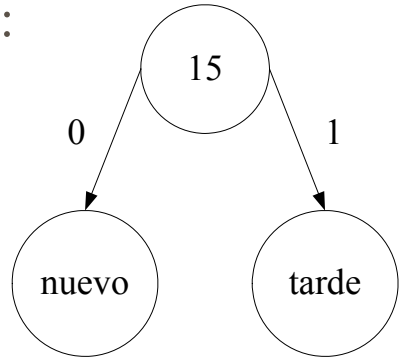
(4, 5, 7, 8, 12, 29)

Ejemplo del Código de Huffman (cont.)

- Luego se eligen los dos valores más pequeños y se construye un árbol binario con hojas etiquetadas:

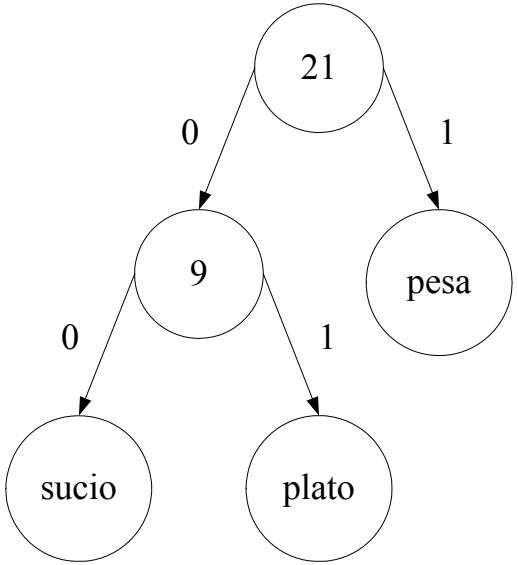


- Se reemplazan los dos valores por su suma, obteniéndose una nueva secuencia (7, 8, 9, 12, 29). De nuevo, se toman los dos valores más pequeños y se construye el árbol binario:



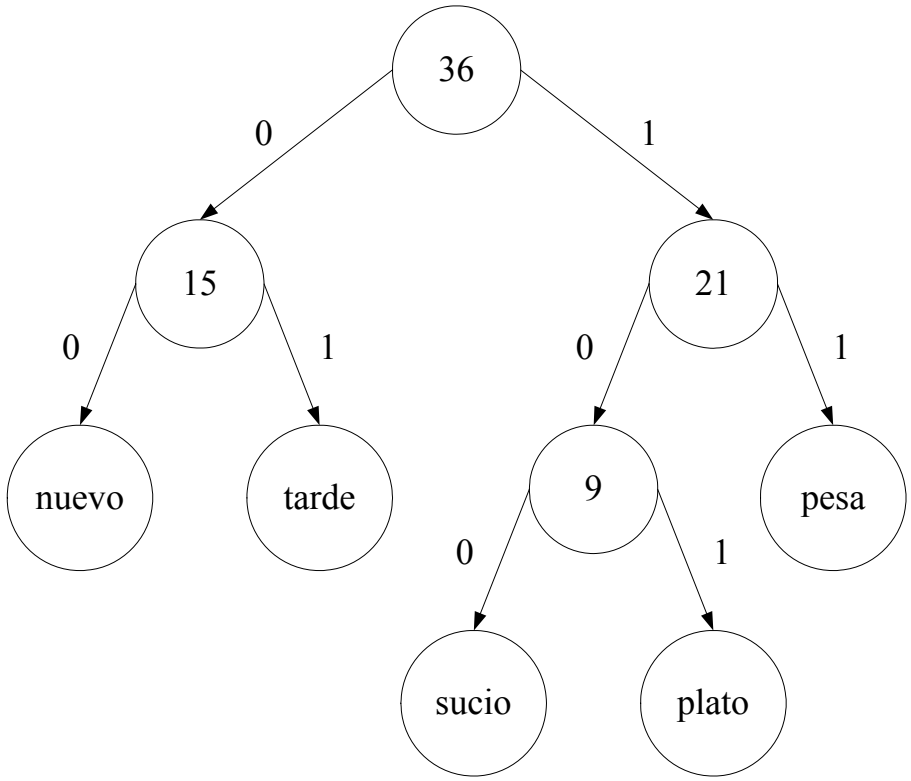
Ejemplo del Código de Huffman (cont.)

- Ahora se tienen las frecuencias (9, 12, 15, 29) y una vez más se seleccionan las menores:



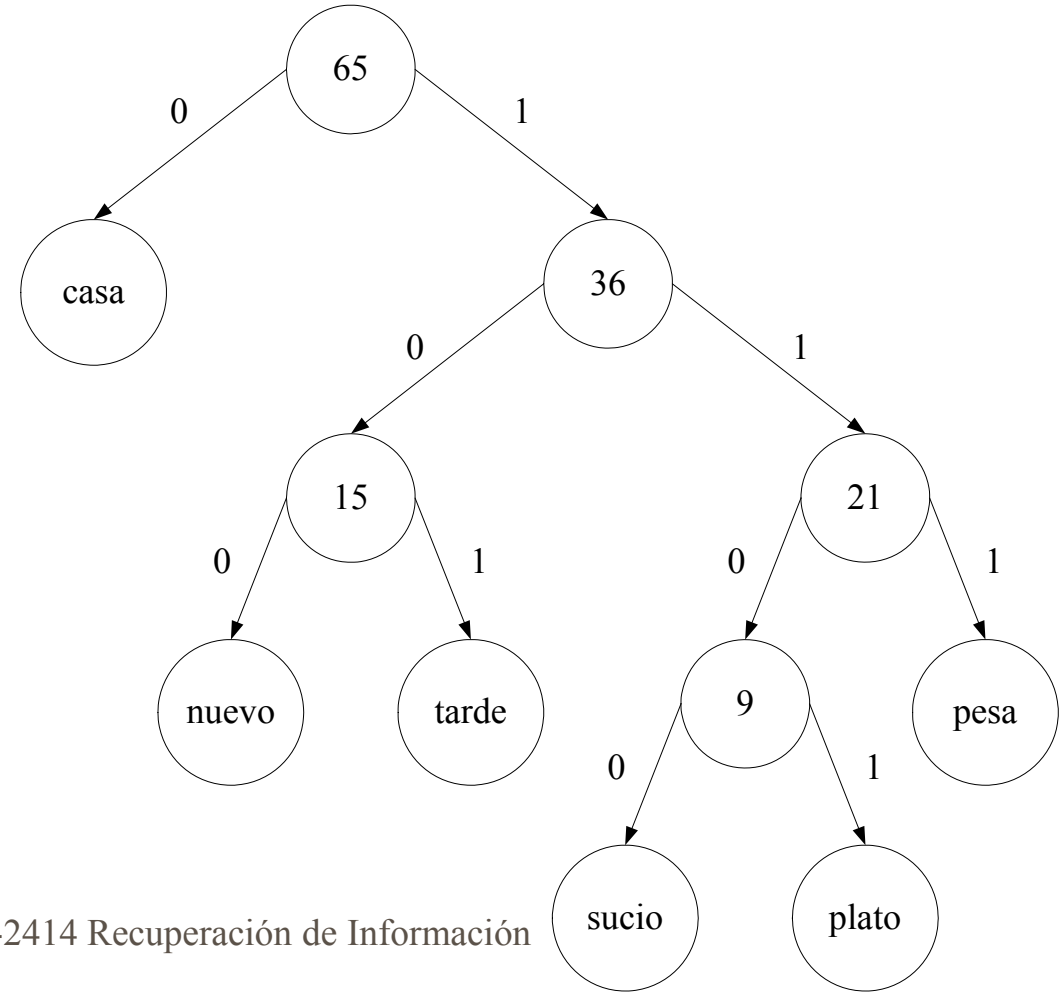
Ejemplo del Código de Huffman (cont.)

- Ahora se tienen las frecuencias (15, 21, 29) y una vez más se seleccionan las menores:



Ejemplo del Código de Huffman (cont.)

- Las dos frecuencias restantes, 29 y 36, se combinan en el árbol final:



Ejemplo del Código de Huffman (cont.)

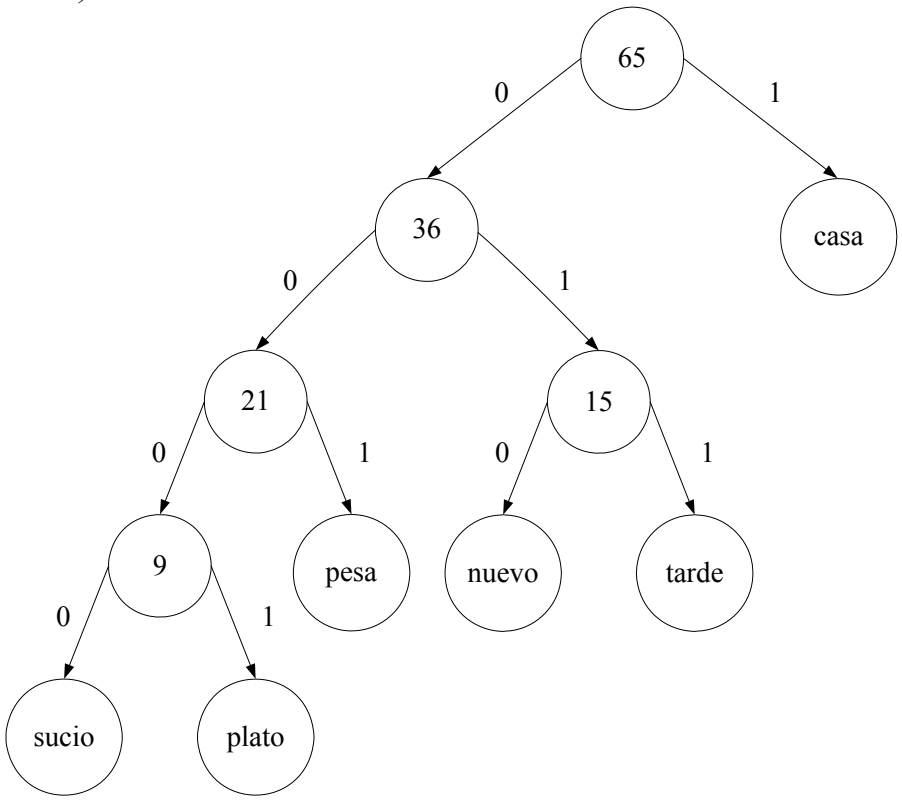
- Del árbol anterior obtenemos el código para este alfabeto:

casa	0
nuevo	100
pesa	111
plato	1101
sucio	1100
tarde	101

- Sustituimos cada palabra del texto por el código respectivo y, una vez hecho esto, agrupamos los bits en grupos de ocho, es decir en bytes.

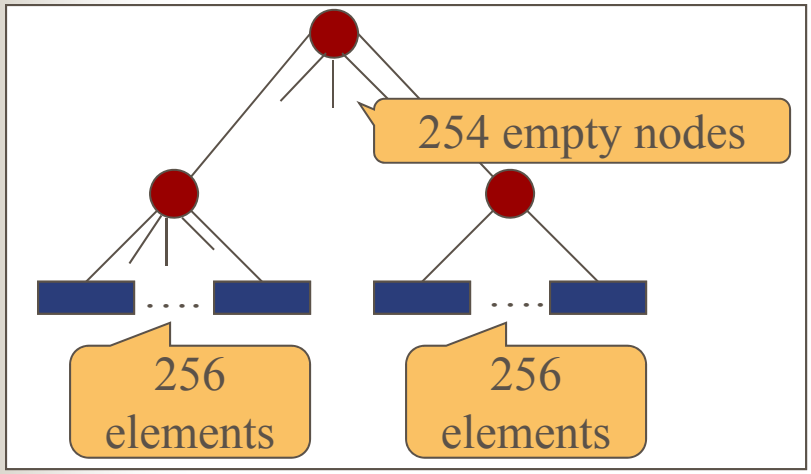
Código de Huffman (cont.)

- *Árbol de Huffman Canónico:* La altura (profundidad) del subárbol izquierdo de cualquier nodo es más grande o igual que la altura (profundidad) del subárbol derecho de ese nodo.

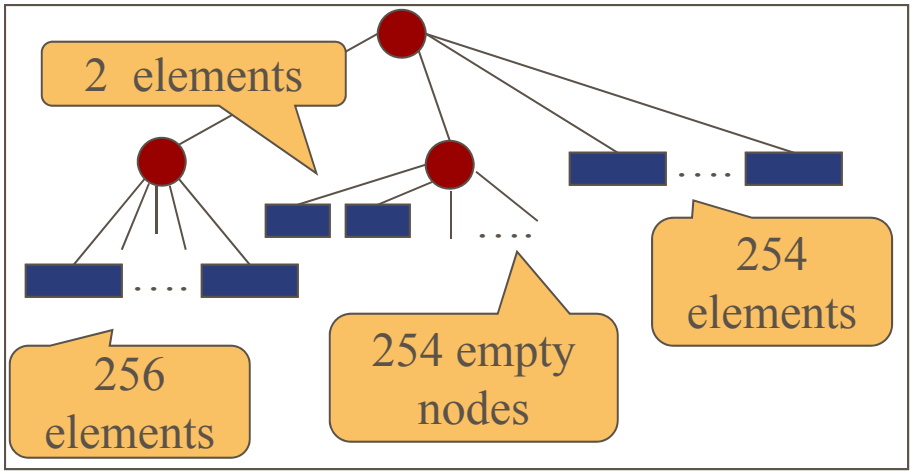


Código de Huffman (cont.)

- *Código de Huffman Orientado a Bytes:*
  - Código asignado a cada palabra del texto es una sucesión de bytes enteros
  - Tiene grado 256 (=2^8), en lugar de 2.



Árbol no óptimo



Árbol óptimo

Código de Huffman (cont.)

- Características del Código de Huffman Orientado a Bytes:
  - Más rápido porque los cambios y las operaciones de enmascarando de los bits no son necesarios.
  - Ninguna disminución significante de la proporción de compresión es experimentada cuando los símbolos son palabras.
  - La búsqueda directa en el texto comprimido es posible.

Pros y Contras de la Compresión

- Pros:
  - Se obtienen mejoras en la velocidad de transmisión.
  - Reduce gastos de espacio, reduce la cantidad de bytes del texto.
  - Reduce sobrecarga de operaciones de entrada y salida.
- Contras:
  - Se debe almacenar la información relativa a la codificación, por lo que para textos cortos no obtendremos mucha reducción de tamaño.
  - Tiempo de codificación y descodificación.
  - Algunos algoritmos no permiten el acceso aleatorio del texto.

Referencias Bibliográficas

- La información fue tomada de:
  - Libro de texto del curso.
  - Presentaciones del curso RI del estudiante Marlon Campos. Universidad de Costa Rica, 2003.