Curso
2010 | 2011

Compresión de Huffman

Práctica 4

David Morales Sáez
Alberto Manuel Mireles Suárez

Introducción

En esta práctica hemos tenido que analizar el funcionamiento del código de compresión de Huffman con el fin de codificar un texto de manera que logremos reducir el tamaño del mismo.

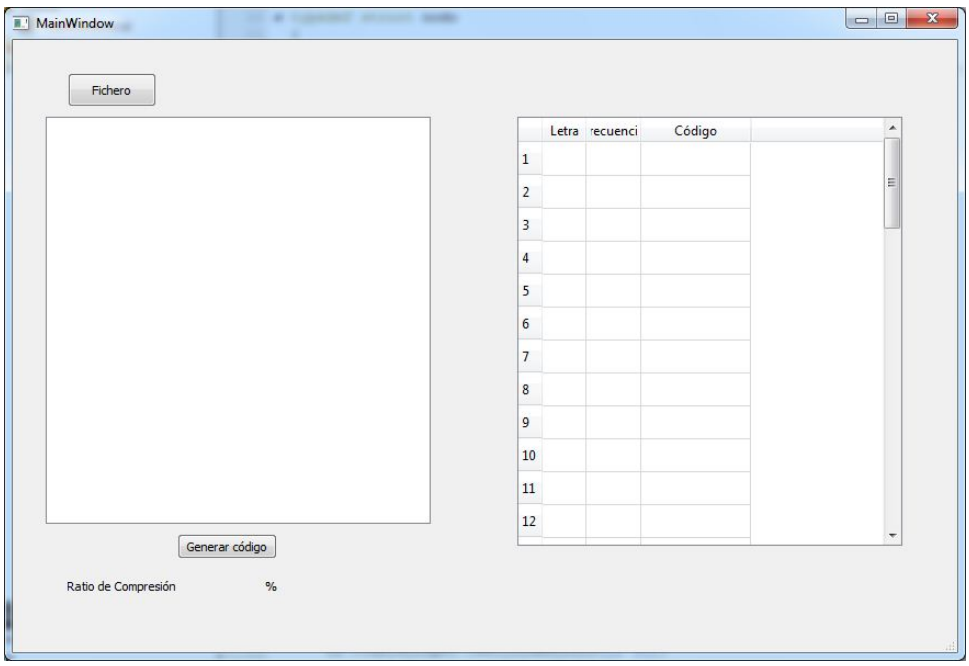
Desarrollo de la práctica

Para lograr comprimir un texto cualquiera, hemos de emplear algún tipo de codificación. En este caso el código de Huffman. Se trata de un código que permite una mayor compresión cuanto menor sea el número de caracteres diferentes en el texto, debido a que el tamaño de las palabras código aumenta conforme más caracteres hayan.

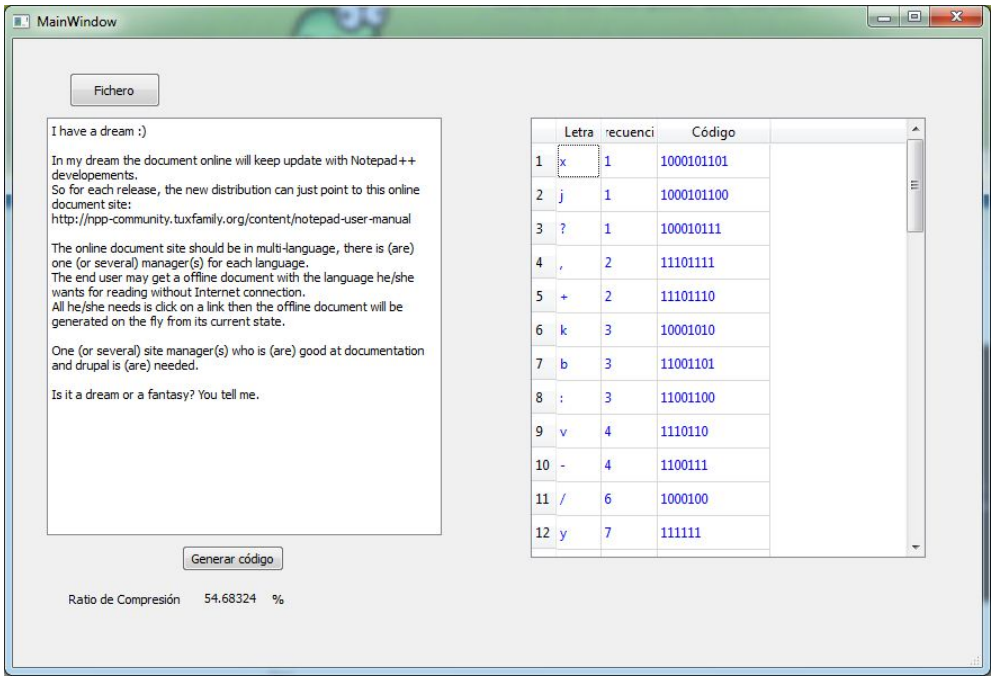
Para realizar la codificación, primero examinamos el texto para obtener la frecuencia de aparición de cada carácter para asociar a los caracteres más frecuentes las palabras-código más cortas. Luego, para obtener el código de Huffman se crea un árbol binario formado por nodos que contienen un carácter y su frecuencia, de modo que a cada rama se le asigna un 1 o un 0. Para obtener el código binario del carácter seguimos el camino adecuado a través del árbol, añadiendo un 0 si es una rama 0, o un 1 si es una rama 1.

Implementación

Para realizar la práctica hemos creado una interfaz gráfica en la que podemos introducir un texto manualmente o seleccionar un archivo guardado en memoria para su compresión.



Una vez cargado el texto, al pulsar el botón “Generar Código” se generará el código de compresión correspondiente y a su vez mostrará el ratio de compresión logrado.



Conclusión

El ratio de codificación de los códigos de Huffman para los textos es bastante aceptable, aunque tenemos que tener en cuenta que existen una serie de factores que nos permiten explotar al máximo sus ventajas:

- Para textos demasiado cortos no generan una gran ventaja, y puede ocurrir que la compresión ocupe mayor espacio puesto que tenemos que almacenar la información del código para permitir su descompresión.

- El código es más efectivo cuanto menor sea el número de caracteres diferentes en el texto. Esto se debe a que el número de bits de los caracteres menos frecuentes aumenta rápidamente.
- El código es más efectivo conforme mayor sea la diferencia de probabilidades entre los caracteres más y menos frecuentes, ya que los caracteres más frecuentes usan menos bits que los menos frecuentes.