COMP261 Tutorial Week 10

Data Encoding and Decoding

Data encoding and decoding are techniques used to convert data from one form to another. Encoding refers to the process of converting data into a format that can be easily transmitted, stored, or processed, while decoding is the reverse process of converting encoded data back to its original form.

There are many encoding and decoding techniques available, each with its own strengths and weaknesses. In this tutorial, we will explore some common techniques used for data encoding and decoding, including fixed-length encoding, variable-length encoding, and prefix encoding.

- ## Fixed-length encoding

Here is an example of fix-length encoding and decoding for 26 letters, period (.), comma (,), and quotation mark ("):

Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code
A	00000	J	01001	S	10010	,	11011
B	00001	K	01010	T	10011	"	11100
C	00010	L	01011	U	10100	space	11101
D	00011	M	01100	V	10101		
E	00100	N	01101	W	10110		
F	00101	O	01110	X	10111		
G	00110	P	01111	Y	11000		
H	00111	Q	10000	Z	11001		
I	01000	R	10001	.	11010		

1. *Can you encode “HELLO WORLD” using the above encoding scheme?*
2. *Using this encoding method, what is the compression rate achieved?*

   *(The string "hello world" is now represented using 55 bits: 11 characters * 5 bits per character. The original ASCII code uses 11 * 8 = 88 bits).*
3. *If you received the following binary sequence: “10110000100001101000”, how do you decode it?*
4. *How do we decide the number of bits of our code in the codebook?*
5. *What is potential advantage and disadvantage of the fixed-length method? (Difficulty to encode/decode, compression rate, generalizability, …)*

- ## Variable-length encoding

1. Morse code:

Morse code works by representing each letter and number as a unique sequence of dots and dashes, which are also known as dits and dahs. Each dot represents a short sound or light, and each dash represents a longer sound or light. The length of a dash is three times the length of a dot. The sequence of dits and dahs is used to spell out words and phrases.

For example, the letter A is represented by a single dit followed by a single dah, while the letter B is represented by a single dah followed by three dits. The letter C is represented by a single dah followed by a single dit followed by a single dah, and so on.

One of the reasons for the popularity of Morse code is its versatility. Because it uses a simple system of dots and dashes, it can be used to transmit messages across a wide range of mediums, including sound, light, and electricity. This has made it a valuable tool for everything from military communications to amateur radio operators and even hobbyists.

Here is the Morse codebook:

A	.-	F	..-.	K	-.-	P	.--.
B	-...	G	--.	L	.-..	Q	--.-
C	-.-.	H	....	M	--	R	.-.
D	-..	I	..	N	-.	S	...

E	.	J	.---	O	---	T	-
U	..-	V	...-	W	.--	X	-..-
Y	-.--	Z	--..	1	.----	2	..---
3	...--	4	....-	5	.....	6	-....
7	--...	8	---..	9	----.	0	-----

In Morse code, the length of the code for each letter is determined by its relative frequency of use in the English language. More commonly used letters are assigned shorter codes, while less commonly used letters are assigned longer codes. This is based on statistical analysis of the frequency of letters in the English language.

1. *If you replace '.' by '0' and '-' by '1', you can get a binary Morse codebook. Now, please encode "Helloworld" and see the compression rate by comparing to the original ASCII code.*

2. *In Morse code, to split each letter when transmitting, the Morse code operator uses brief pauses between the dots and dashes within each letter, and longer pauses between letters to signal the end of one letter and the beginning of the next.*

   Please decode the following sequence you received:

   ```
   000(pause)111(pause)000
   ```

3. *Is it possible to decode a sequence of Morse code without knowing where the pauses are? Suppose we have the following Morse code sequence:*

   ```
   .-..---.--..-.-..-.----
   ```

   ```
   01001110110010100101111
   ```

   Try and see different possible decoding results.

4. *Discussion: Morse code is a simple and efficient encoding scheme for human-readable texts, but why it is not suitable for transmitting large amounts of data over the internet?*

2. Huffman coding:

Huffman coding is a data compression algorithm that uses variable-length codes to represent data. It was invented by David Huffman in 1952 while he was a student at MIT. The basic idea behind Huffman coding is to represent more common data elements with shorter codes, and less common elements with longer codes. This is achieved by constructing a binary tree, called a Huffman tree, based on the frequency of occurrence of each element in the data.

Here's an example of a Huffman coding tree:

```
            +
        /       \
       +          +
    /     \    /     \
   +      C    D      E
  / \
 B   A
```

The codebook is:

A: 001, B: 000, C: 01, D: 10, E: 11

To use a Huffman coding tree to decode a message, you need to traverse the tree starting from the root node and moving down either the left or right branch based on the code you are trying to decode. The code is a sequence of bits, typically represented as 0s and 1s, that corresponds to a particular character in the original message. You start at the root node and examine the first bit of the code. If it is a 0, you move down the left branch; if it is a 1, you move down the right branch. You keep following the branches, examining the next bit of the code until you reach a leaf node. The character represented by that leaf node is the decoded character for that code. You then repeat this process for each code in the message until you have decoded the entire message.

1. *Please decode the following sequence:*

   `11111111101001011001000000001001011011`

2. *Comparing to the fix-length encoding like this:*

   `A: 000, B: 001, C: 010, D: 011, E: 100`

   What is the compression rate?

3. *Do you need extra schemes or codes to split letters in this variable-length method? Why?*

4. *Write seudo-code or Java code for decoding a sequence based on a given Huffman coding tree.*

5. *What are the advantages and disadvantages of Huffman encoding/decoding? (Generalizability? Computational Cost? Redundancy? …)*