Multimedia Information Retrieval and Technology

Lecture 19
Compression Algorithms II

By : Fangyu Wu

Room: SD555

Xi'an Jiaotong-Liverpool University
西交利物浦大学

1. Introduction
2. Variable-Length Coding (VLC)
   - Shannon-Fano Algorithm
   - Huffman Coding Algorithm
3. Basics of Information Theory
4. **LZW Compression**

1. Introduction
2. Variable-Length Coding (VLC)
    - **Shannon-Fano Algorithm**
    - **Huffman Coding Algorithm**
3. Basics of Information Theory
4. LZW Compression
    - Encoder
    - Decoder

LZW Coding

- LZW uses **fixed-length codewords** to represent variable-length strings of symbols/characters that commonly occur together, e.g., words in English text.

- The LZW encoder and decoder build up the same dictionary dynamically while receiving the data.

LZW Coding

- LZW places longer and longer repeated entries (an element) into a dictionary, and then emits the *code* for an element, rather than the string itself, if the element has already been placed in the dictionary.

- LZW algorithm is an **adaptive**, **dictionary based** compression technique.

LZW Coding

ALGORITHM 7.2 LZW COMPRESSION

```
BEGIN
    s = next input character;
    while not EOF
        {
          c = next input character;

          if s + c exists in the dictionary
             s = s + c;
        else
          {
          output the code for s;
          add string s + c to the dictionary with a new code;
          s = c;
          }
        }
    output the code for s;
END
```

Example 7.2 LZW compression for string "ABABBAB-CABABBA"

- Let's start with a very simple **dictionary** (also referred to as a "string table"), initially containing only 3 characters, with codes as follows:

code	string
1	A
2	B
3	C

- Now if the input string is "ABABBABCABABBA", the LZW compression algorithm works as follows:

s	c	output	code	string
			1	A
			2	B
			3	C
A	B	1	4	AB
B	A	2	5	BA
A	B			
AB	B	4	6	ABB
B	A			
BA	B	5	7	BAB

What's next?

Xi'an Jiaotong-Liverpool University
西交利物浦大學

s	c	output	code	string
			1	A
			2	B
			3	C
A	B	1	4	AB
B	A	2	5	BA
A	B			
AB	B	4	6	ABB
B	A			
BA	B	5	7	BAB
B	C	2	8	BC
C	A	3	9	CA
A	B			
AB	A	4	10	ABA
A	B			
AB	B			
ABB	A	6	11	ABBA
A	EOF	1		

LZW Coding

The output codes are 1 2 4 5 2 3 4 6 1. Instead of 14 characters, only 9 codes need to be sent.

If we assume each character or code is transmitted as a byte, the compression ratio would be 14/9 =1.56.

The LZW is an adaptive algorithm, in which the encoder and decoder independently build their own string tables.

No overhead involving transmitting the string table.

LZW Coding

The LZW assigns fixed length code words to variable length sequences of source symbols.

Integrated into a variety of mainstream imaging file formats.

1. Introduction
2. Variable-Length Coding (VLC)
    - **Shannon-Fano Algorithm**
    - **Huffman Coding Algorithm**
    - **Adaptive Huffman Coding Algorithm**
3. Basics of Information Theory
4. LZW Compression
    - Encoder
    - Decoder

LZW DECOMPRESSION

```
BEGIN
    s = NIL;
    while not EOF
        {
          k = next input code;
          entry = dictionary entry for k;
          output entry;
          if (s != NIL)
             add string s + entry[0] to dictionary
             with a new code;
          s = entry;
        }
END
```

EXAMPLE 7.3 LZW decompression for string ABABBABCABABBA

Input codes to the decoder are 1 2 4 5 2 3 4 6 1. The initial string table is identical to what is used by the encoder.

The LZW decompression algorithm then works as follows:

s	k	entry/output	code	string
			1	A
			2	B
			3	C
NIL	1			A

s	k	entry/output	code	string
			1	A
			2	B
			3	C
NIL	1	A		
A	2	B	4	AB
B	4	AB	5	BA
AB	5	BA	6	ABB
BA	2	B	7	BAB
B	3	C	8	BC
C	4	AB	9	CA
AB	6	ABB	10	ABA
ABB	1	A	11	ABBA
A	EOF			

Apparently the output string is ABABBABCABABBA — a truly lossless result!

Exercise II

Consider the dictionary-based LZW compression algorithm. Suppose the alphabet is the set of symbols {0, 1}. Show the dictionary (symbol sets plus associated codes) and output for LZW compression of the input

0 1 1 0 0 1 1

Solution

Answer:
With input 0 1 1 0 0 1 1, we have

				DICTIONARY	
w	k	wk	output	index	symbol
-	-	--	------	-----	------
NIL	0	0			
0	1	01	0	2	01
1	1	11	1	3	11
1	0	10	1	4	10
0	0	00	0	5	00
0	1	01			
01	1	011	2	6	011