1. Introduction > 1.1 Introduction / Overview

Please provide the introduction / overview on this lesson

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

Overview

In this chapter, you are going to learn about:

- Introduction to **LZ78**
- How to make **encoder** and **decoder** of LZ78
- **Examples** on encoder and decoder of LZ78

1. Introduction > 1.2 Learning Content

Please make sure the hierarch of the content is well formed.
Please organize the lesson in 3-5 main topics and use 3-level headings.

Level 1	Level 2	Level 3
1. Concept of LZ78	1.1 Introduction to LZ78	
	1.2 Algorithm of LZ78	
	1.3 Encoder	
	1.4 Decoder	
2. Examples		

1. Introduction > 1.3 Learning Content

ID Will do it by looking at 1.1 Lesson overview

Image Processing	
I. General knowledge in image p rocessing and multimedia	1. Introduction to Image Processing 2. Data Structure and Color of Images 3. Ms. Visual Studio 2008 and OpenCV 4. Introduction to Multimedia Systems 5. Introduction to Video and Lossless Compression 6. Huffman Coding 7. LZ77 8. LZ78 9. LZW
II. Advance knowledge in image segmentation and luminance	10. Sampling 11. Image Segmentation-I 12. Image Segmentation-II 13. Luminance and Histogram Equalization

Please provide objective of the lesson by high light keyword and follow (Audience, Behavior, Condition, Degree) to write the objective

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

Objective
Upon completion of this chapter, you will be able to: • Understand concept of **LZ78** • Use method of **compression** and **decompression** of LZ78 for applying to text.

1. Introduction > 1.5 Keywords

Please provide keywords of the lesson with explanation

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

Keywords	Description
Compression utility	is a software program that is used to **compress** or **decompress** files.
Index	is an **alphabetical list** of names, subjects, etc. with reference to the pages on which they are mentioned.
Address	is the coded representation of the **physical** or **logical** location of a source or destination resource, such as a register, a memory partition, an application, or a node or station.

2. Learn> Topic: 1.1 Introduction to LZ78

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- Q: What is **LZ78**?
- A: **LZ78** is the lossless data compression algorithm.
- It was published by Abraham **L**empel and Jacob **Z**iv in 19**78**.
- The most popular modification of LZ78 is **LZW** made by Terry Welche.
- LZ78 has **high** requirements on **space** because the dictionary can occupy the whole free memory.
- But, there are several ways to solve this problem.
  - If we run out of memory, we can freeze the dictionary or delete the whole dictionary and begin to make new one.
  - In the UNIX utility **compress**, it freezes of the dictionary and monitors the compression ratio.
    - If this ratio falls below the predefined threshold, the entire dictionary is deleted and a new one is being created.

2. Learn> Topic: 1.2 Algorithm of LZ78

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- LZ78 is based on a dictionary that will be created dynamically at runtime.

- Both the encoding and the decoding process use the same rules to ensure that an identical dictionary is available.
    - This dictionary contains any sequence already used to build the former contents.
    - The compressed data have the general form:
        - Index addressing an entry of the dictionary
        - First beginning symbol

- In contrast to LZ77 no combination of address and sequence length is used.
    - Instead only the index to the dictionary is stored.

2. Learn> Topic: 1.2 Algorithm of LZ78

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- LZ78-based schemes work by entering phrases into a dictionary.
- Then when a repeat occurrence of that particular phrase is found
  - outputting the dictionary index instead of the phrase.

- Every step LZ78 will send **a pair (*i*, *a*)** to the output.
  - where ***i*** is an index of the phrase into the dictionary.
  - and ***a*** is the next symbol following immediately after the found phrase.

- In each step we look for the longest phrase in dictionary, that would correspond to the unprocessed part of the input text.
  - Index of this phrase together with the symbol, which follows the found part in input text, are then send to the output.
  - The old phrase extended by the new symbol is then put into dictionary.

2. Learn> Topic: 1.2 Algorithm of LZ78

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- Q: What is **encoder** algorithm of LZ78?
- A: **Encoder** algorithm of LZ78:
  1) Find all **different symbols** from an input string or a file.
  2) Build a base dictionary which contains **entries** and **codewords**.
  3) Build an output dictionary which contains **entries**, **codewords**, and **outputs.**
  4) Start to do loop from the **first symbol**.
  5) If a match is **found** from the base dictionary, add another symbol and compare again.
  6) If a match is **not found** from the base dictionary, output codeword of those symbol (form: <?,?>).
  7) Repeat to do loop from next matching symbol until the last symbol or **E**nd **O**f **F**ile (**EOF**).

2. Learn> Topic: 1.2 Algorithm of LZ78

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- **Pseudocode** of encoder:

```
begin
   initialize a dictionary by empty phrase P
   while (not EOF) do
   begin
         readSymbol(X)
         if (F.X is in the dictionary) then
            F = F.X
         else
         begin
           output(pointer(F),X)
           encode X to the dictionary
           initialize phrase F by empty character
         end
   end
end
```

2. Learn> Topic: 1.2 Algorithm of LZ78

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- Q: What is **decoder** algorithm of LZ78?
- A: **Decoder** algorithm of LZ78:
  1) Use the base dictionary from **encoder**.
  2) Use result of encoder as **input**.
  3) Build an output dictionary which contains **inputs**, **codewords**, **entries**, and **outputs** (entries must exactly the same as outputs).
  4) Start to do loop for **finding outputs** by using the input one by one.
  5) Repeat loop until **E**nd **O**f **F**ile.

2. Learn> Topic: 1.2 Algorithm of LZ78

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- **Pseudocode** of decoder:

```
begin
   initialize a dictionary by empty phrase
   while (not EOF) do
   begin
         read pair of index and character (i,X) from input
         put new phrase (i,X) into dictionary
         generate phrase to the output
   end
end
```

2. Learn> Topic: 1.3 Encoder

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- Q: If we have an input string: “abcaacdaabcdabbbab”. What is encode of this string?
- A: First, we find all different characters.
  - There are “a, b, c, d”.

- Then build a base dictionary.

Entry	Codeword
a	1
b	2
c	3
d	4

2. Learn> Topic: 1.3 Encoder

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- Input string: “abcaacdaabcdabbbab”.
- Next, build an output dictionary.
  - ➢ a b c a a c d a a b c d a b b b a b EOF
    ⬆
  - ➢ Start from “a”.
    - ○ “a” has in the base dictionary
    - ○ So, we take “ab”.
    - ○ “ab” doesn’t have in the dictionary.
    - ○ We add “ab” to output dictionary.
    - ○ The last codeword is 4.
    - ○ New codeword must be 5.

Entry	Codeword	Output
ab	5	<1,2>

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

2. Learn> Topic: 1.3 Encoder

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- a b c a a c d a a b c d a b b b a b EOF
- Now we are at "c".
  - ○ "c" has in the base dictionary
  - ○ So, we take "ca".
  - ○ "ca" doesn't have in the both dictionary.
  - ○ We add "ca" to output dictionary.
  - ○ The last codeword is 5.
  - ○ New codeword must be 6.

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

Entry	Codeword	Output
ab	5	<1,2>
ca	6	<3,1>

2. Learn> Topic: 1.3 Encoder

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- a b c a a c d a a b c d a b b b a b EOF
- Now we are at “a”.
  - “a” has in the base dictionary
  - So, we take “ac” (“ac” ≠ “ca”).
  - “ac” doesn’t have in the both dictionary.
  - We add “ac” to output dictionary.
  - The last codeword is 6.
  - New codeword must be 7.

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

Entry	Codeword	Output
ab	5	<1,2>
ca	6	<3,1>
ac	7	<1,3>

2. Learn> Topic: 1.3 Encoder

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- a b c a a c d a a b c d a b b b a b EOF
  ⬆

- Now we are at “d”.
  - ○ “d” has in the base dictionary
  - ○ So, we take “da”.
  - ○ “da” doesn’t have in the both dictionary.
  - ○ We add “da” to output dictionary.
  - ○ The last codeword is 7.
  - ○ New codeword must be 8.

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

Entry	Codeword	Output
ab	5	<1,2>
ca	6	<3,1>
ac	7	<1,3>
da	8	<4,1>

2. Learn> Topic: 1.3 Encoder

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- a b c a a c d a a b c d a b b b a b EOF
  ⬆

- Now we are at “a”.
  - ○ “a” has in the base dictionary
  - ○ So, we take “ab”.
  - ○ “ab” has in the output dictionary.
  - ○ We take “abc”.
  - ○ “abc” doesn’t have in the both dictionary.
  - ○ We add “abc” to output dictionary.

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

Entry	Codeword	Output
ab	5	<1,2>
ca	6	<3,1>
ac	7	<1,3>
da	8	<4,1>
abc	9	<5,3>

2. Learn> Topic: 1.3 Encoder

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- a b c a a c d a a b c d a b b b a b EOF

  ⬆ (under the second "d")

- Now we are at “d”.
  - “d” has in the base dictionary
  - So, we take “da”.
  - “da” has in the output dictionary.
  - We take “dab”.
  - “dab” doesn’t have in the both dictionary.
  - We add “dab” to output dictionary.

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

Entry	Codeword	Output
ab	5	<1,2>
ca	6	<3,1>
ac	7	<1,3>
da	8	<4,1>
abc	9	<5,3>
dab	10	<8,2>

2. Learn> Topic: 1.3 Encoder

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- a b c a a c d a a b c d a b b b a b EOF
  ⬆

- Now we are at “b”.
  - ○ “b” has in the base dictionary
  - ○ So, we take “bb”.
  - ○ “bb” doesn’t have in the both dictionary.
  - ○ We add “bb” to output dictionary.

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

Entry	Codeword	Output
ab	5	<1,2>
ca	6	<3,1>
ac	7	<1,3>
da	8	<4,1>
abc	9	<5,3>
dab	10	<8,2>
bb	11	<2,2>

2. Learn> Topic: 1.3 Encoder

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- a b c a a c d a a b c d a b b b a b EOF
  ⬆

- Now we are at “a”.
  - ○ “a” has in the base dictionary → “ab”.
  - ○ But, there is no next character (EOF).
  - ○ We just add codeword to “Output”.

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

Entry	Codeword	Output
ab	5	<1,2>
ca	6	<3,1>
ac	7	<1,3>
da	8	<4,1>
abc	9	<5,3>
dab	10	<8,2>
bb	11	<2,2>
EOF	EOF	<5, >

Encoder = {<1,2>, <3,1>, <1,3>, <4,1>, <5,3>, <8,2>, <2,2>, <5, >}

2. Learn> Topic: 1.4 Decoder

	☒ A : Text-based + Audio ☐ B : Text-based + Video ☐ C : Only Video
(1) Learning Contents	see below

- Input string is the result of encoder.
  - string = {<1,2>, <3,1>, <1,3>, <4,1>, <5,3>, <8,2>, <2,2>, <5, >, EOF}

- Q: How to make decoder?
- A: First, use the base dictionary from encoder.
- Next, build an output dictionary.

Input	Codeword	Entry	Output

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

2. Learn> Topic: 1.4 Decoder

(1) Learning Contents

☒ A : Text-based ＋ Audio
☐ B : Text-based ＋ Video
☐ C : Only Video

- Input string is the result of encoder.
  - string = {<1,2>, <3,1>, <1,3>, <4,1>, <5,3>, <8,2>, <2,2>, <5, >, EOF}
- Output dictionary:
  - Start “<1,2>”.
  - Codeword: “1” is “a” and “2” is “b”.
  - Entry is “ab”. So, output is also “ab”.
  - New codeword is 5.

Input	Codeword	Entry	Output
<1,2>	5	ab	ab

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

2. Learn> Topic: 1.4 Decoder

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- Input string is the result of encoder.
  - string = {<1,2>, <3,1>, <1,3>, <4,1>, <5,3>, <8,2>, <2,2>, <5, >, EOF}

    ⬆

- Output dictionary:
  - Next “<3,1>”.
  - Codeword: “3” is “c” and “1” is “a”.
  - Entry is “ca”. So, output is also “ca”.
  - New codeword is 6.

Input	Codeword	Entry	Output
<1,2>	5	ab	ab
<3,1>	6	ca	ca

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

2. Learn> Topic: 1.4 Decoder

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- Input string is the result of encoder.
  - string = {<1,2>, <3,1>, <1,3>, <4,1>, <5,3>, <8,2>, <2,2>, <5, >, EOF}

⬆

- Output dictionary:
  - Next "<1,3>".
  - Codeword: "1" is "a" and "3" is "c".
  - Entry is "ac". So, output is also "ac".
  - New codeword is 7.

Input	Codeword	Entry	Output
<1,2>	5	ab	ab
<3,1>	6	ca	ca
<1,3>	7	ac	ac

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

2. Learn> Topic: 1.4 Decoder

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- Input string is the result of encoder.
  - string = {<1,2>, <3,1>, <1,3>, <4,1>, <5,3>, <8,2>, <2,2>, <5, >, EOF}
- Output dictionary:
  - Next "<4,1>".
  - Codeword: "4" is "d" and "1" is "a".
  - Entry is "da". So, output is also "da".
  - New codeword is 8.

Input	Codeword	Entry	Output
<1,2>	5	ab	ab
<3,1>	6	ca	ca
<1,3>	7	ac	ac
<4,1>	8	da	da

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

2. Learn> Topic: 1.4 Decoder

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- Input string is the result of encoder.
  - string = {<1,2>, <3,1>, <1,3>, <4,1>, <5,3>, <8,2>, <2,2>, <5, >, EOF}
- Output dictionary:
  - Next “<5,3>”.
  - Codeword: “5” is “ab” and “1” is “a”.
  - Entry is “aba”. So, output is also “aba”.
  - New codeword is 9.

Input	Codeword	Entry	Output
<1,2>	5	ab	ab
<3,1>	6	ca	ca
<1,3>	7	ac	ac
<4,1>	8	da	da
<5,3>	9	aba	abc

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

2. Learn> Topic: 1.4 Decoder

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- Input string is the result of encoder.
  - string = {<1,2>, <3,1>, <1,3>, <4,1>, <5,3>, <8,2>, <2,2>, <5, >, EOF}

- Output dictionary:
  - Next "<8,2>".
  - Codeword: "8" is "da" and "2" is "b".
  - Entry is "dab". So, output is also "dab".
  - New codeword is 10.

Input	Codeword	Entry	Output
<1,2>	5	ab	ab
<3,1>	6	ca	ca
<1,3>	7	ac	ac
<4,1>	8	da	da
<5,3>	9	aba	abc
<8,2>	10	dab	dab

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

2. Learn> Topic: 1.4 Decoder

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- Input string is the result of encoder.
  - string = {<1,2>, <3,1>, <1,3>, <4,1>, <5,3>, <8,2>, <2,2>, <5, >, EOF}
- Output dictionary:
  - Next “<2,2>”.
  - Codeword: “2” is “b”.
  - Entry is “bb”. So, output is also “bb”.
  - New codeword is 11.

Input	Codeword	Entry	Output
<1,2>	5	ab	ab
<3,1>	6	ca	ca
<1,3>	7	ac	ac
<4,1>	8	da	da
<5,3>	9	aba	abc
<8,2>	10	dab	dab
<2,2>	11	bb	bb

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

2. Learn> Topic: 1.4 Decoder

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- Input string is the result of encoder.
  - string = {<1,2>, <3,1>, <1,3>, <4,1>, <5,3>, <8,2>, <2,2>, <5, >, EOF}

- Output dictionary:
  - Last "<5, >".
  - Codeword: "5" is "ab".
  - Entry is "ab". So, output is also "ab".

Input	Codeword	Entry	Output
<1,2>	5	ab	ab
<3,1>	6	ca	ca
<1,3>	7	ac	ac
<4,1>	8	da	da
<5,3>	9	aba	abc
<8,2>	10	dab	dab
<2,2>	11	bb	bb
<5, >	EOF	ab	ab

Entry	Codeword
a	1
b	2
c	3
d	4

Base dictionary

Decoder is:

"abcaacdaabcdabbbab"

2. Learn> Topic: 2. Examples

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- Example1: Input string: “bcaebaedbcac”. Find encoder and decoder of LZ78?
- **Encoder:** First, we find all different characters.
  - ➢ There are “a, b, c, d, e”.

- Then build a base dictionary.

Entry	Codeword
a	1
b	2
c	3
d	4
e	5

2. Learn> Topic: 2. Examples

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- Input string: “bcaebaedbcac”.
- Next, build an output dictionary.
  - ➢ b c a e b a e d b c a c EOF
    ⬆
  - ➢ Start from “b”.
    - ○ “b” has in the base dictionary
    - ○ So, we take “bc”.
    - ○ “bc” doesn’t have in the dictionary.
    - ○ We add “bc” to output dictionary.
    - ○ The last codeword is 5.
    - ○ New codeword must be 6.

Entry	Codeword	Output
bc	6	<2,3>

Entry	Codeword
a	1
b	2
c	3
d	4
e	5

Base dictionary

2. Learn> Topic: 2. Examples

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- b c a e b a e d b c a c EOF
  ⬆
- Now we are at “a”.
  - ○ “a” has in the base dictionary
  - ○ So, we take “ae”.
  - ○ “ae” doesn’t have in the dictionary.
  - ○ We add “ae” to output dictionary.
  - ○ The last codeword is 6.
  - ○ New codeword must be 7.

Entry	Codeword	Output
bc	6	<2,3>
ae	7	<1,5>

Entry	Codeword
a	1
b	2
c	3
d	4
e	5

Base dictionary

2. Learn> Topic: 2. Examples

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- ➢ b c a e b a e d b c a c EOF

  ⬆

- ➢ Now we are at “b” (seen until EOF).
  - ○ “b” has in the base dictionary
  - ○ So, we take “ba”.
  - ○ “ba” doesn’t have in the dictionary.
  - ○ We add “ba” to output dictionary.

Entry	Codeword	Output
bc	6	<2,3>
ae	7	<1,5>
ba	8	<2,1>
ed	9	<5,4>
bca	10	<6,1>
EOF	EOF	<3, >

Entry	Codeword
a	1
b	2
c	3
d	4
e	5

Base dictionary

Encoder = {<2,3>, <1,5>, <2,1>, <5,4>, <6,1>, <3, >}

2. Learn> Topic: 2. Examples

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- **Decoder:** Input string is the result of encoder.
  - string = {<2,3>, <1,5>, <2,1>, <5,4>, <6,1>, <3, >, EOF}

- First, use the base dictionary from encoder.
- Next, build an output dictionary.

Input	Codeword	Entry	Output

Entry	Codeword
a	1
b	2
c	3
d	4
e	5

Base dictionary

2. Learn> Topic: 2. Examples

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

(1) Learning Contents

- **Decoder:** Input string is the result of encoder.
  - string = {<2,3>, <1,5>, <2,1>, <5,4>, <6,1>, <3, >, EOF}

- Output dictionary:
  - Start “<2,3>”.
  - Codeword: “2” is “b” and “3” is “c”.
  - Entry is “bc”. So, output is also “bc”.
  - New codeword is 6.

Input	Codeword	Entry	Output
<2,3>	6	bc	bc

Entry	Codeword
a	1
b	2
c	3
d	4
e	5

Base dictionary

2. Learn> Topic: 2. Examples

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- **Decoder:** Input string is the result of encoder.
  - string = {<2,3>, <1,5>, <2,1>, <5,4>, <6,1>, <3, >, EOF}

- Output dictionary:
  - Next "<1,5>" (see until EOF).
  - Codeword: "1" is "a" and "5" is "e".
  - Entry is "ae". So, output is also "ae".

Input	Codeword	Entry	Output
<2,3>	6	bc	bc
<1,5>	7	ae	ae
<2,1>	8	ba	ba
<5,4>	9	ed	ed
<6,1>	10	bca	bca
<3, >	EOF	c	c

Decoder is: "bcaebaedbcac"

Entry	Codeword
a	1
b	2
c	3
d	4
e	5

Base dictionary

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

(1) Learning Contents

- Example2: Input string: “dadadacdadfee”. Find encoder and decoder of LZ78?
- **Encoder:** First, we find all different characters.
  - There are “a, b, c, d, e, f”.
- Then build a base dictionary.

Entry	Codeword
a	1
b	2
c	3
d	4
e	5
f	6

2. Learn> Topic: 2. Examples

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- Input string: “dadadacdadfee”.
- Next, build an output dictionary.
  - ➢ d a d a d a c d a d f e e EOF
    ⬆
  - ➢ Start from “d”.
    - ○ “d” has in the base dictionary
    - ○ So, we take “da”.
    - ○ “da” doesn’t have in the dictionary.
    - ○ We add “da” to output dictionary.
    - ○ The last codeword is 6.
    - ○ New codeword must be 7.

Entry	Codeword	Output
da	7	<4,1>

Entry	Codeword
a	1
b	2
c	3
d	4
e	5
f	6

Base dictionary

(1) Learning Contents

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

- d a d a d a c d a d f e e EOF
  ⬆

- Now we are at "d".
  - ○ "d" has in the base dictionary
  - ○ So, we take "da".
  - ○ "da" has in the output dictionary.
  - ○ We take "dad".
  - ○ "dad" doesn't have in the dictionary.
  - ○ We add "dad" to output dictionary.
  - ○ The last codeword is 7.
  - ○ New codeword must be 8.

Entry	Codeword	Output
da	7	<4,1>
dad	8	<7,4>

Entry	Codeword
a	1
b	2
c	3
d	4
e	5
f	6

Base dictionary

2. Learn> Topic: 2. Examples

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- d a d a d a c d a d f e e EOF

  ⬆

- Now we are at “a” (seen until EOF).
  - ○ “a” has in the base dictionary
  - ○ So, we take “ac”.
  - ○ “ac” doesn’t have in the dictionary.
  - ○ We add “ac” to output dictionary.

Entry	Codeword	Output
da	7	<4,1>
dad	8	<7,4>
ac	9	<1,3>
dadf	10	<8,6>
ee	11	<5,5>
EOF		

Entry	Codeword
a	1
b	2
c	3
d	4
e	5
f	6

Base dictionary

Encoder = {<4,1>, <7,4>, <1,3>, <8,6>, <5,5>}

2. Learn> Topic: 2. Examples

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

(1) Learning Contents

- **Decoder:** Input string is the result of encoder.
  - string = {<4,1>, <7,4>, <1,3>, <8,6>, <5,5>, EOF}

- First, use the base dictionary from encoder.
- Next, build an output dictionary.

Input	Codeword	Entry	Output

Entry	Codeword
a	1
b	2
c	3
d	4
e	5
f	6

Base dictionary

2. Learn> Topic: 2. Examples

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- **Decoder:** Input string is the result of encoder.
  - string = {<4,1>, <7,4>, <1,3>, <8,6>, <5,5>, EOF}
    ⬆

- Output dictionary:
  - Start “<4,1>”.
  - Codeword: “4” is “d” and “1” is “a”.
  - Entry is “da”. So, output is also “da”.
  - New codeword is 7.

Input	Codeword	Entry	Output
<4,1>	7	da	da

Entry	Codeword
a	1
b	2
c	3
d	4
e	5
f	6

Base dictionary

2. Learn> Topic: 2. Examples

(1) Learning Contents

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

- **Decoder:** Input string is the result of encoder.
  - string = {<4,1>, <7,4>, <1,3>, <8,6>, <5,5>, EOF}

    ⬆

- Output dictionary:
  - Next "<7,4>" (see until EOF).
  - Codeword: "7" is "da" and "4" is "d".
  - Entry is "dad". So, output is also "dad".

Input	Codeword	Entry	Output
<4,1>	7	da	da
<7,4>	8	dad	dad
<1,3>	9	ac	ac
<8,6>	10	dadf	dadf
<5,5>	11	ee	ee
EOF			

Decoder is: "dadadacdadfee"

Entry	Codeword
a	1
b	2
c	3
d	4
e	5
f	6

Base dictionary

4. Outro > 4.1 Summarize

Please give a lesson summary.
Each topic can be summarized into a sentence, diagram, or even a word.

☒ A : Text-based + Audio
☐ B : Text-based + Video
☐ C : Only Video

Summarize

- **LZ78** is the lossless data compression algorithm. It has high requirements on space because the dictionary can occupy the whole free memory.

- **Encoder** algorithm of LZ78:
    1) Find all different symbols from an input string or a file.
    2) Build a base dictionary which contains entries and codewords.
    3) Build an output dictionary which contains entries, codewords, and outputs.
    4) Start to do loop from the first symbol.
    5) If a match is found from the base dictionary, add another symbol and compare again.
    6) If a match is not found from the base dictionary, output codeword of those symbol (form: <?,?>).
    7) Repeat to do loop from next matching symbol until the last symbol or End Of File (EOF).

4. Outro > 4.1 Summarize

Please give a lesson summary.
Each topic can be summarized into a sentence, diagram, or even a word.

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

Summarize (cont.)

- **Decoder** algorithm of LZ78:
  1) Use the base dictionary from encoder.
  2) Use result of encoder as input.
  3) Build an output dictionary which contains inputs, codewords, entries, and outputs (entries must exactly the same as outputs).
  4) Start to do loop for finding outputs by using the input one by one.
  5) Repeat loop until End Of File.

4. Outro > 4.2 References

Provide references if you think the students need.

Reference

- https://en.wikipedia.org/wiki/LZ77_and_LZ78
- http://www.binaryessence.com/dct/en000140.htm
- http://www.stringology.org/DataCompression/lz78/index_en.html

4. Outro > 4.3 Assignment

Please provide the assignment such as exercise , discussion, research topic, Short essay, case studies, ...

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

Assignment

1) Find encoder and decoder of LZ78? If we have:

- ➢ Input string: “abdcaedbdcecabbdeacb”

2) Find encoder and decoder of LZ78? If we have:

- ➢ Input string: “cdaabbefacbdegfeeabfedegg”

4. Outro > 4.4 Next Lesson

This is the end of the lesson.
Ending message and introduction to next lesson including lesson title and topics should be given.

☒ A : Text-based + Audio
□ B : Text-based + Video
□ C : Only Video

Overview

- Introduce to LZW
- Concept of LZW

Next Lesson Title	**LZW** 1. Concept of LZW 2. Examples