# 한글정보전송에 있어서 Huffman 부호화의 一方案

°이 종 헌* 신 승 호** 진 용 옥*

* 경희대학교 전자공학과, ** 인천대학 전자계산학과

( A proposal idea of Huffman coding in Hangul message transmission )

Jong Heon Lee* Seung Ho Shin** Yong Ohk Chin*

* Dept. of Elec. Eng., Kyunghee Univ. **Dept. of Comp. Eng., Incheon College

ABSTRACT

This paper proposed using Huffman codes in order to reduce the average codes length on aspect of Hangul transmission. When Hangul is parted into three independent information source and each of these sources is coded by Huffman codes, the average number of bits need only 10.308 bits to compose of one character.

We execute the simulation which converts between Huffman codes and ASCII codes corresponding to hangul phoneme.

## 1. 서 론

정보전송의 문제에 있어서 통신로 용량( channel capacity) C가 유한하기때문에 C 보다 높은 비율로 정보원( information source) 를 에러없이 전송할 수는 없다. [1,2,3] 따라서 전송 속도를 높이기 위해서는 C 를 증가시켜야 하지만 경비가 많이 소요되어 적절한 정보원 부호화( source coding) 를 고려하게 된다. 즉, 보다 짧은 평균길이의 부호로 정보원을 부호화하여 전송한다면 통신로 용량이 증가된것과 같은 효과를 기대할 수 있다.

이러한 관점에서 가변길이 부호인 Huffman 부호로 한국어 정보원을 부호화하는 방법에대하여 고찰한 것이다. 현재 사용되고있는 ASCII 부호를 Huffman 부호로 변환하는 것을 포함한 Huffman 부호의 이용 가능성에 대하여 고찰하였다.

## 2. 한국어 정보원

자연언어는 구성의 특성상 random process가 되며 [6], 이에 대한 해석은 통계적인 확률분포에 따르게 된다. C.E.Shannon 에 의해 정보이론의 기초가 확립된 이후 각종언어에 대한 정보이론적인 결과가 발표된바 있으며[4], 한글에 대해서도 이러한 관점의 연구와[5,6,7] 한글 모오스 부호의 개선에 대하여 발표된바 있다.[6]

한국어 정보원은 정보원을 구성하는 요소에 따라 대략 다음과 같이 2가지로 분류할 수 있다.

{1} . 기본문자 (24개) + 단어 space (1 개)=25개
{2} . 기본문자 (24개) + 앙자음 (5 개) + 복자음 (11 개) + 복모음(11 개) + 단어space=52개

{1} 의 경우 entropy 는 4.0769(bits) 이고 [6,7] , {2} 의 경우에는 4.3319(bits) 이다.[8]

에러를 고려하지 않았을 경우에 대한 Shannon 의 부호화 정리에 의하면 부호의 평균길이 L 은 {1},{2}의 각각에 대해

$$4.0769 \le \overline{L} < 5.0769$$

$$4.3319 \le \overline{L} < 5.3319$$

이다. 또 한 글자를 전송하는데 필요한 요소의 평균 갯수는 space 를 포함시켰을 때 {1} 의 경우에는 3.03 개 [8], {2} 의 경우에는 2.69 개 [8] 이므로 한 글자를 전송하는데 필요한 평균 비트수 $\overline{L_1}$ 의 상한과 하한은 각각

$$12.35 \le \overline{L_1} < 15.38$$

$$11.65 \le \overline{L_1} < 14.34$$

이다.

또 실제로 정보원 {1},{2} 에 대해서 Huffman 부호화했을때 평균 부호길이는 각각 4.1152(bits),

4.3377(bits)이다. [8]

본 논문에서는 한 글자당 소요되는 평균 비트수를 줄이기위해 다음과 같이 한글을 특별한 3개의 정보원으로 분리하였다.

S1). 초성(19) + 단어 space =20개
S2). 중성(21)
S3). 종성(27) + Null =28개
( Null: 종성이 없는 경우)

구체적인 통계자료는 참고문헌 [5]에서 구하였다. 단어 space는 어디에 포함시켜도 되며 종성에 Null을 포함시켜 특별히 첨가된 부호없이 초, 중, 종성을 구분하여 모아쓰기 처리를 할 수 있도록 하였다.

표 1,2,3에 의하면 정보원 S1,S2,S3의 entropy와 평균 부호길이 $\bar{L}$ 의 상한과 하한은 각각

$$H(S1)=3.37199 \ , \quad 3.37199 \le \bar{L} < 4.37199$$
$$H(S2)=3.41851 \ , \quad 3.41851 \le \bar{L} < 4.41851 \qquad (1)$$
$$H(S3)=2.28556 \ , \quad 2.28556 \le \bar{L} < 3.28556$$

이다. 따라서 식(1) 에서 주어진 $\bar{L}$ 의 상한과 하한 내에서 정보원 S1,S2,S3를 부호화할 수 있다. 또 한 글자를 전송하는데 필요한 평균 비트수는

$$10.207 \le \bar{L} < 13.530$$

으로 정보원을 (1),(2)로 구성할 때보다 줄아질 수 있다.

## 3.Huffman 부호

가변길이 부호는 문자그대로 길이가 일정치 않으므로 일의복호가능(uniquely decodable)하고 순시복호가능(instantaneous decodable) 하지 않으면 직렬전송된 부호를 복호할 수가 없다. 일의복호가능하고 순시복호가능하기 위해서는 어떤부호어의 어두(prefix)가 다른 부호어로 사용되면 안된다. [1,2] Kraft 와 McMillan은 다음 부등식이 radix r인 부호의 일의복호가능하고 순시복호가능하기 위한 필요충분조건임을 증명하였다.[1,2]

$$\sum_{i=1}^{M} r^{-l_i} \le 1 \qquad (2)$$

여기서 M 은 부호어의 갯수이며 $l_i$ 는 각 부호의 길이이다.

Huffman 은 식(2) 의 조건을 만족하는 최단길이부호를 만드는 방법을 고안하였으며 이를 Huffman 부호라 한다.[1,2] 표 1,2,3은 각각 정보원 S1,S2,S3에 대한 확률분포와 그에 따른 Huffman 부호이다.

표 1. 정보원 S1의 확률 분포와 Huffman 부호
Table 1. Probability distribution and Huffman codes of source S1

| 요 소 | 확 률 | $-p\log_2 p$ | Huffman 부호 |
|---|---|---|---|
| space | 0.25101 | 0.50056 | 11 |
| ㅇ | 0.18999 | 0.45522 | 100 |
| ㄱ | 0.10692 | 0.33349 | 010 |
| ㅅ | 0.06247 | 0.24993 | 1011 |
| ㄷ | 0.06247 | 0.24993 | 0000 |
| ㅎ | 0.05963 | 0.24258 | 0001 |
| ㅈ | 0.05806 | 0.23841 | 0010 |
| ㄹ | 0.05399 | 0.22736 | 0011 |
| ㄴ | 0.05103 | 0.21906 | 0110 |
| ㅁ | 0.03449 | 0.16755 | 10100 |
| ㅂ | 0.02855 | 0.14648 | 01110 |
| ㅊ | 0.01294 | 0.08115 | 101011 |
| ㅌ | 0.00817 | 0.05666 | 011111 |
| ㅈ | 0.00735 | 0.05490 | 1010100 |
| ㄸ | 0.00600 | 0.04430 | 0111100 |
| ㄲ | 0.00459 | 0.03592 | 0111101 |
| ㅋ | 0.00374 | 0.03017 | 10101011 |
| ㅆ | 0.00192 | 0.01735 | 101010101 |
| ㅃ | 0.00132 | 0.01266 | 1010101000 |
| ㅉ | 0.00084 | 0.00861 | 1010101001 |
| 계 | 0.99978 | 3.37199 | |

표 2. 정보원 S2의 확률 분포와 Huffman 부호
Table 2. Probability distribution and Huffman codes of source S2

| 요소 | 확 률 | $-p\log_2 p$ | Huffman 부호 |
|---|---|---|---|
| ㅏ | 0.21618 | 0.47769 | 01 |
| ㅣ | 0.15255 | 0.41382 | 001 |
| ㅡ | 0.02781 | 0.14372 | 010 |
| ㅓ | 0.10241 | 0.33668 | 110 |
| ㅗ | 0.09369 | 0.32004 | 111 |
| ㅜ | 0.06343 | 0.25236 | 0110 |
| ㅔ | 0.04691 | 0.20707 | 00000 |
| ㅋ | 0.04469 | 0.20038 | 00001 |
| ㅐ | 0.04099 | 0.18890 | 00011 |
| ㅢ | 0.02781 | 0.14372 | 01111 |
| ㅘ | 0.02076 | 0.11606 | 000101 |
| ㅟ | 0.01612 | 0.08677 | 011101 |
| ㅛ | 0.00892 | 0.06076 | 0111000 |

| | | | |
|---|---|---|---|
| ㅠ | 0.00717 | 0.05108 | 0111001 |
| ㅑ | 0.00665 | 0.04811 | 00010000 |
| ㅋ | 0.00564 | 0.04212 | 00010001 |
| ㅐ | 0.00462 | 0.03590 | 00010011 |
| ㅓ | 0.00435 | 0.03415 | 000111100 |
| ㅒ | 0.00049 | 0.00540 | 0001001010 |
| ㅖ | 0.00012 | 0.00161 | 00010010110 |
| ㅔ | 0.00008 | 0.00107 | 00010010111 |
| 계 | 0.99999 | 3.41851 | |

표 3. 정보원 S3의 확률 분포와 Huffman 부호
Table 3. Probability distribution and Huffman codes of source S3

| 요 소 | 확 률 | $-p\log_2 p$ | Huffman 부호 |
|---|---|---|---|
| NULL | 0.55701 | 0.47024 | 0 |
| ㄴ | 0.14290 | 0.40114 | 100 |
| ㄹ | 0.08450 | 0.30122 | 111 |
| ㅇ | 0.06612 | 0.25911 | 1010 |
| ㄱ | 0.04645 | 0.20569 | 1100 |
| ㅁ | 0.02653 | 0.13892 | 10111 |
| ㅆ | 0.02271 | 0.12402 | 11011 |
| ㅂ | 0.01851 | 0.10654 | 101100 |
| ㅅ | 0.01504 | 0.09107 | 101101 |
| ㅄ | 0.00313 | 0.02604 | 1101011 |
| ㅌ | 0.00312 | 0.02600 | 11010000 |
| ㄶ | 0.00274 | 0.02335 | 11010100 |
| ㄷ | 0.00208 | 0.01855 | 11010101 |
| ㅎ | 0.00200 | 0.01791 | 110100010 |
| ㅍ | 0.00185 | 0.01682 | 110100001 |
| ㅈ | 0.00136 | 0.01300 | 110100110 |
| ㅊ | 0.00130 | 0.01243 | 110100111 |
| ㄺ | 0.00063 | 0.00669 | 1101001010 |
| ㄲ | 0.00054 | 0.00590 | 11010010000 |
| ㄵ | 0.00053 | 0.00577 | 11010010001 |
| ㄼ | 0.00037 | 0.00425 | 11010010010 |
| ㄻ | 0.00036 | 0.00412 | 11010010011 |
| ㄾ | 0.00030 | 0.00346 | 11010010110 |
| ㄳ | 0.00014 | 0.00176 | 110100101111 |
| ㄽ | 0.00004 | 0.00058 | 1101001011101 |
| ㅋ | 0.00004 | 0.00057 | 11010010111000 |
| ㄿ | 0.00003 | 0.00040 | 1101001011100101 |
| ㄿ | 0.000007 | 0.00011 | 1101001011100111 |
| 계 | 1.00003 | 2.28556 | |

여기서 S1,S2,S3에 대한 부호의 평균길이 $\overline{Le1}$,$\overline{Le2}$,$\overline{Le3}$ 는

$$\overline{Le1} = 3.3926 \quad (bits)$$
$$\overline{Le2} = 3.4482 \quad (bits)$$
$$\overline{Le3} = 2.3120 \quad (bits)$$

로 식(1) 을 만족한다. 또 식(2) 의 Kraft 와 McMillan 부등식도

$$S1 : \sum_{i=1}^{32} 2^{-l_i} = 1$$
$$S2 : \sum_{i=1}^{32} 2^{-l_i} = 1$$
$$S3 : \sum_{i=1}^{32} 2^{-l_i} = 1$$

로 만족하며 Huffman 부호가 최단길이 부호라는 것을 알 수 있다. space 를 포함시켰을때 한 글자를 전송하는데 필요한 평균 비트수는 10.308 ( bits) 이다.

## 4. 부호 변환 (code conversion)

표 1,2,3의 Huffman 부호를 이용하여 한글을 전송하기 위해서는 한글에 대응되는 ASCII 부호와 Huffman부호사이에 변환( conversion) 이 필요 하다. 그림 1은 ASCII 부호를 Huffma부호로 변환하는 과정의 흐름도 이고 그림 2는 반대로 Huffman 부호를 ASCII 부호로 바꾸는 과정을 나타낸 것이다.

그림 1,2 에서 알 수 있듯이 종성이 없는 경우 ( Null) 를 하나의 요소로 간주하여 Huffman 부호화 하였기 때문에 수신측에서 모아쓰기 처리가 간단하게 이루어 질 수 있다.

## 5. 실 험

그림 1,2의 흐름도를 프로그램으로 구성하고 키보드로부터 입력된 한글을 RS-232C 를 이용하여 전송하였다. 송신측에서 한글은 해당되는 ASCII 부호로 입력되며 입력된 음소가 조성의 "ㅏ" 인 경우 ASCII 부호로는 "1010010" 이다. 이때 Huffman 부호로는 "010" 이므로 8bit memory에 저장되어 있는 경우 MSB 부터 전송하여고할때 부호의 시작이 어느곳인지 알 수가 없다. 이러한 문제를 해결하기 위해 부호의 MSB 앞에 "1"를 삽가하여 "1010" 으로 저장한 다음 왼쪽으로 shift 시킨다. 최초로 "1"이 carry flag에 shift 되어 들어 오면 그다음 bit 부터 부호가 시작되는 것으로 판단한다.

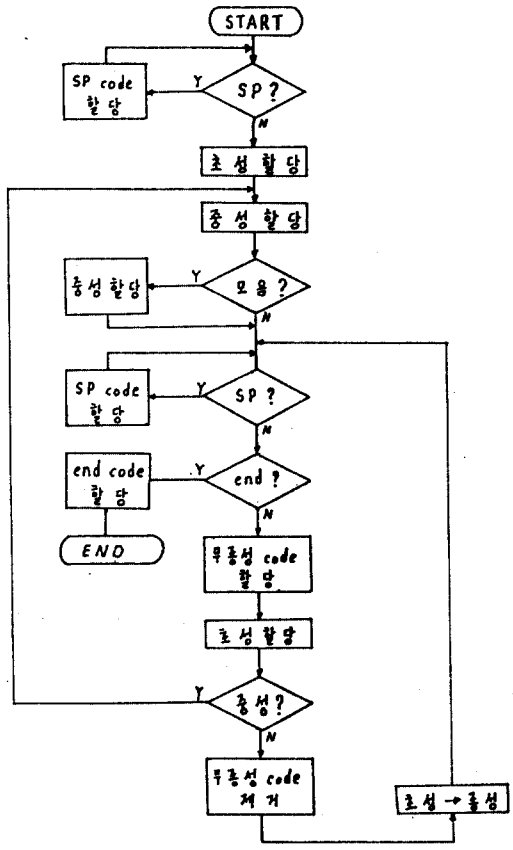


그림 1. ASCII 부호를 Huffman 부호로 변환하는 흐름도

Fig 1. Flow chart of ASCII to Huffman code conversion

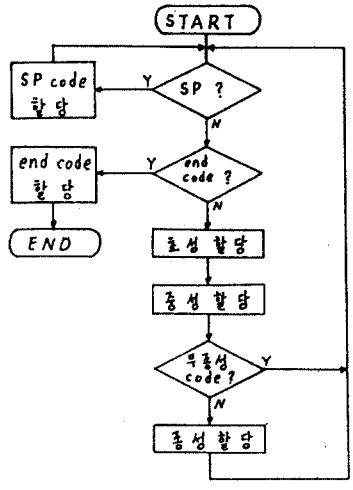


그림 2. Huffman 부호를 ASCII 부호로 변환하는 흐름도

Fig 2. Flow chart of Huffman to ASCII code conversion

이러한 과정을 그림 3에 나타내었다.

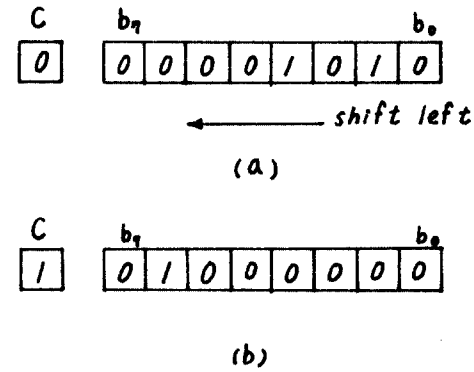


그림 3. (a)Shift시키기 전상태 (b)Shift를 5회 한 후의 상태

Fig 3. (a)Before shift (b)After five time shift

그림3 의 (b)에서 5번 연속으로 shift 한후에 carry flag가 set 되었으므로 그이후의 3bit 가 전송하여야 할 Huffman 부호이다.
이와 같은 과정으로

" 경희대학교 전자공학과"

를

" rudgmlaogkrry wjswkrhdgkrrhk"

로 전송하였다. 수신측에서는 초, 중, 종성이 구분되어 모아쓰기가 가능한지를 확인하기 위하여 각 글자와 글자 사이에 ","를 삽입하도록 program 하였다. 이때 수신된 결과는

" rud,gml,ao,gkr,ry wjs,wk,rhd,gkr,rhk"

로 전송한 message 가 완전하게 수신되었다.

## 6. 결 론

표 1, 2, 3에 의하면 space 를 포함하여 한 글자를 전송하는데 필요한 평균 비트수는 10.308 (bits)이다. 한글에 대응하는 ASCII 부호는 상자음과 복모음중 ㅖ, ㅒ, ㅙ, ㅞ를 포함하므로 space 를 포함했을때 한 글자를 전송하는데 필요한 요소의 갯수는 참고문헌[5] 의 통계자료에 의해 2.86 개 임을 구할 수 있다. 따라서 ASCII 부호를 그대로 전송한다고 할 경우 한 글자를 전송하는데 20.02( bits) 가 필요하다.

이와같이 Huffman 부호를 이용하면 통신로 용량을 증가시키지 않고 전송비율을 향상시킬수 있다. Huffman 부호의 장점으로 다음과 같은점을 들 수 있다.

1.channel 을 사용하는 시간이 단축되므로 통신요율을 저감시킬 수 있다.
2.사용되는 Huffman 부호에 적합한 복호기가 있어야만 정보를 해독할 수 있으므로 부분적으로 encryption이 가능하다.

## * 참 고 문 헌 *


1. R.W. Hamming, "Coding and information theory", Prentice-Hall, 1980.
2. N.Abramson, "Information theory and coding", McGraw-Hill, 1963.
3. C.E.Shannon, "Mathematical theory of communication", Bell system Tech.J. pp. 379~423, July 1948.
4. G.A. Barnard III,"Statistical calculation of word entropies for four western languages", IEEE Trans. on information theory , Vol.IT-1, pp. 49~50, 1955.
5. 최 창선 ," 한국어 정보원의 구조분석에 관한 연구", 숭전대학교 대학원 석사학위논문, 1978.
6. 이 주근,"한국어 정보원의 구조분석과 code 개선 ", 대한 전자공학회지 , 제 16 권 2호, pp.1~7 1978.
7. 안 수길 ," 공백소를 포함한 한글 자소발생확률과 엔트로피" , 대한 전자공학회지, 제 17 권, 2 호 pp. 23~28, 1980.
8. 이 종언 , "한글 자소발생확률과 부호화에 관한 연구 ", 정희대학교 대학원 전자공학과 연구논문집 제5 권 1호 ,pp. 95~97 , 1986.