About encoding the dialect that uses the rice code

ライス符号を用いた方言の符号化について

日本大学　文理学部
情報システム解析学科

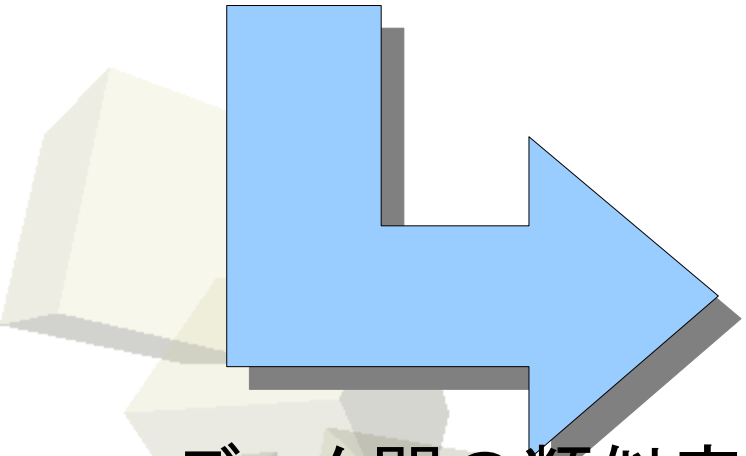
谷研究室　益田真太郎

目次

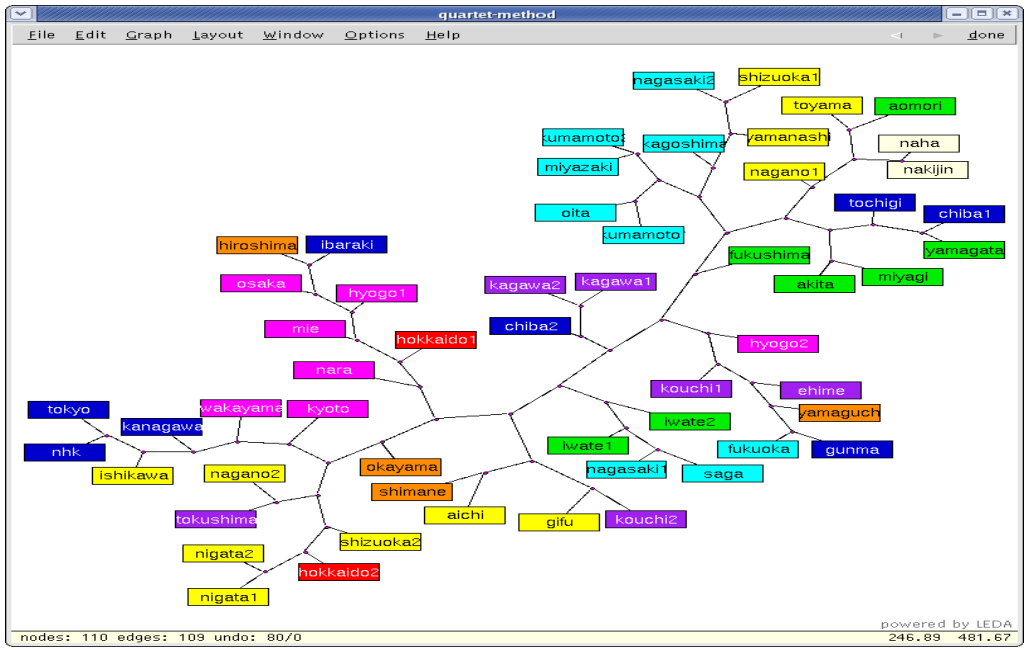
目次

方言の自動分類とは

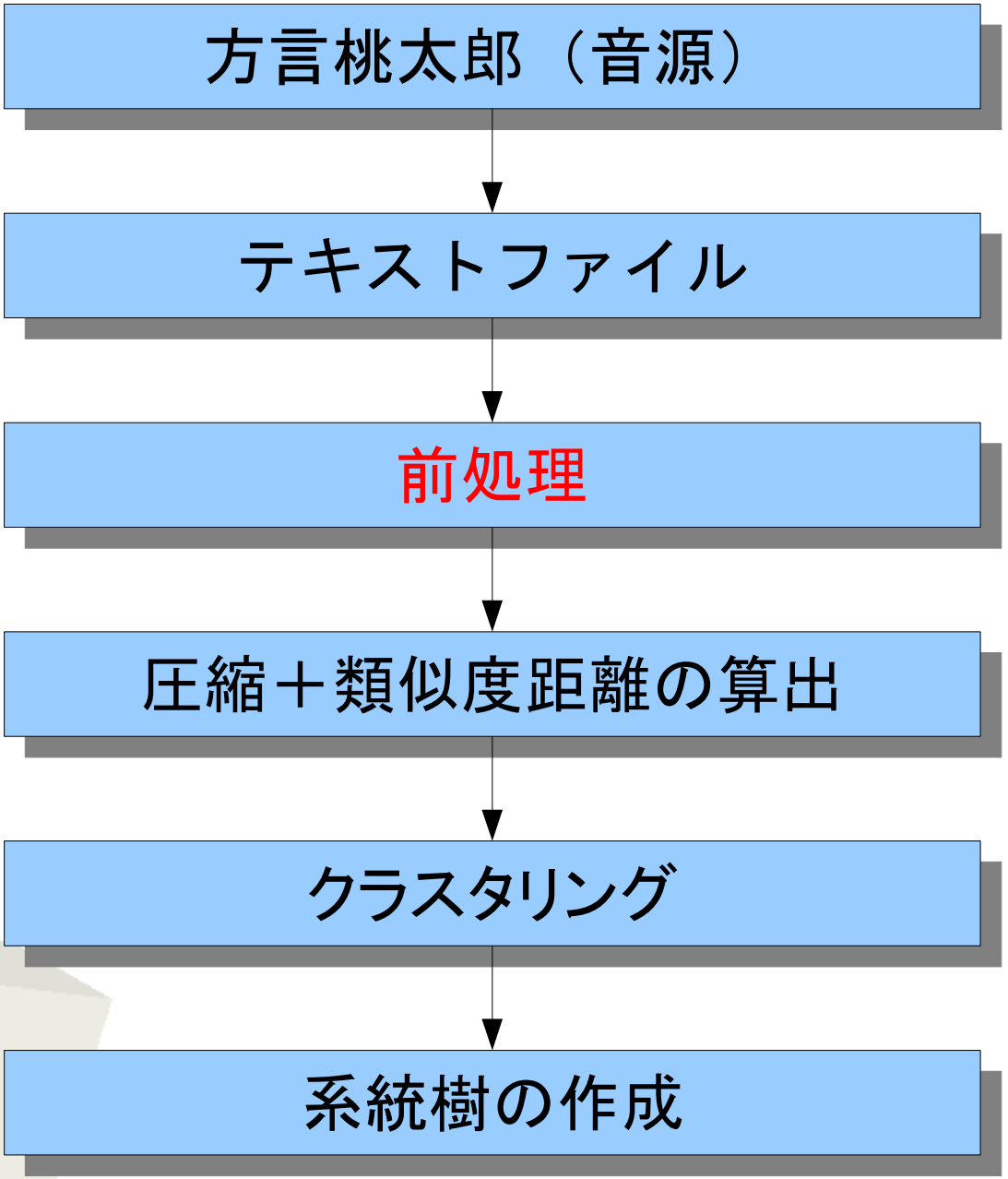
NHK	むかしむかしあるところにおじいさんとおばあさんが…
岩手	むがすむがすあるどころにおずうさんとおばあさんが…
静岡	ずうっとみゃあにあるとこでじいじいとばあばあみ…
大阪	むかしむかしあることろにおじいさんとおばあさんが…
那覇	むかしむかしあるところんかいたんめえとんめえが…



データ間の類似度を
用いて自動分類



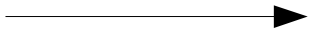
方言の自動分類の手順



前処理とは

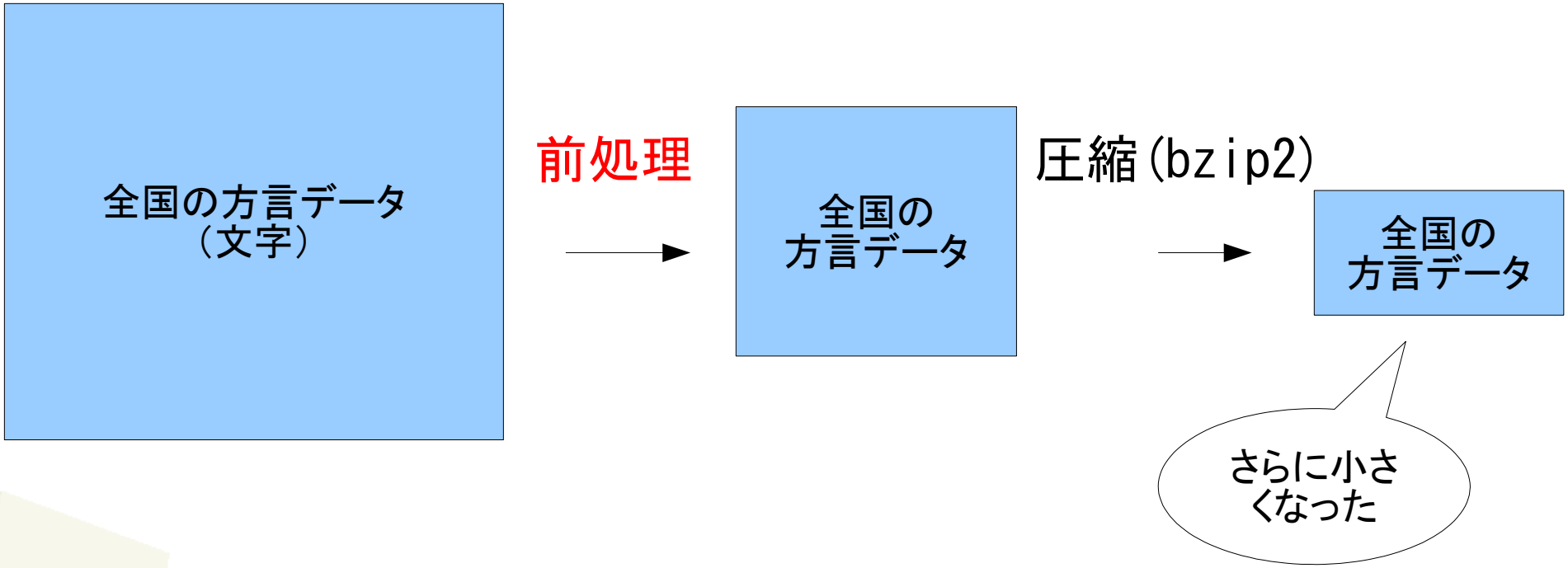
全国の方言データ
（文字）

圧縮（bzip2）



全国の方言データ

前処理とは



前処理とは圧縮前にデータに処理を加えより
圧縮後のデータを小さくすること

なぜ、前処理を行うのか

本研究では先行研究で、良いとされているbzip2(圧縮)を用いて距離を算出する。

圧縮類似度に基づくデータ間の類似度を求める式

$$d(x,y) \approx \frac{bzip2(xy) - bzip2(x)}{bzip2(y)}$$

bzip2が小さいほどより良い圧縮類似度が求まるとされている

なぜ、前処理を行うのか

本研究では先行研究で、良いとされているbzip2(圧縮)を用いて距離を算出する。

圧縮類似度に基づくデータ間の類似度を求める式

$$d(x,y) \approx \frac{bzip2(xy) - bzip2(x)}{bzip2(y)}$$

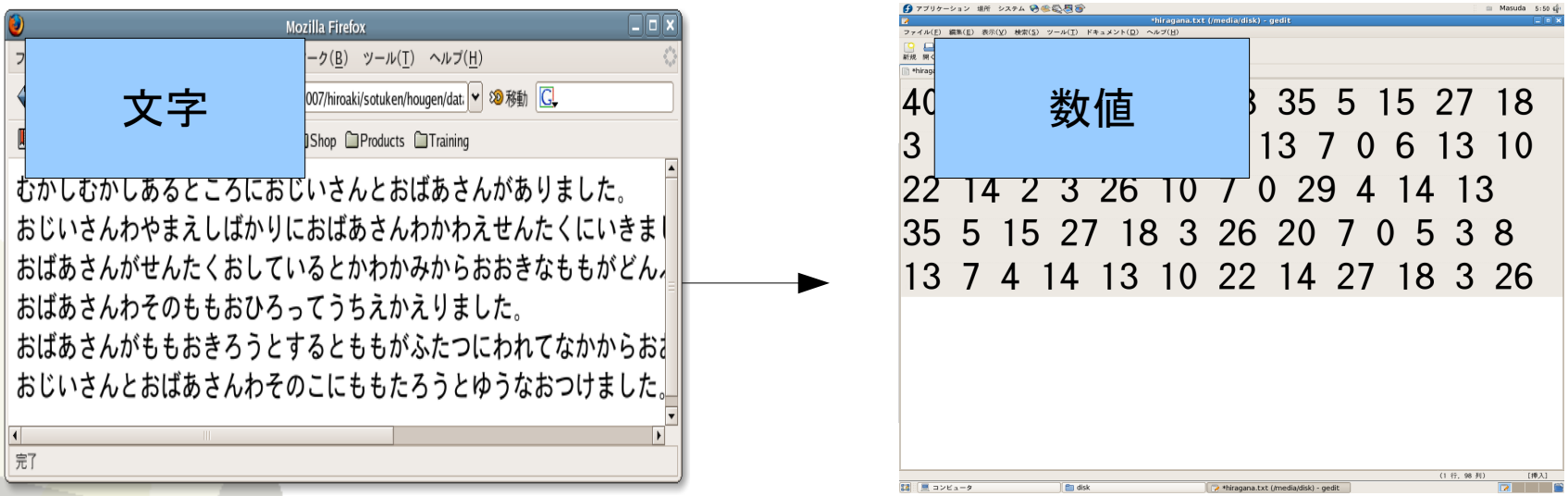
bzip2が小さいほどより良い圧縮類似度が求まるとされている

bzip2(圧縮)をより小さくするため前処理を行う

目次

先行研究での前処理

先行研究では、
圧縮類似度を算出する際には無駄な情報を省くため
文字を数値に置き換えて単純化する。



先行研究では前処理1、2を行った。

先行研究での前処理

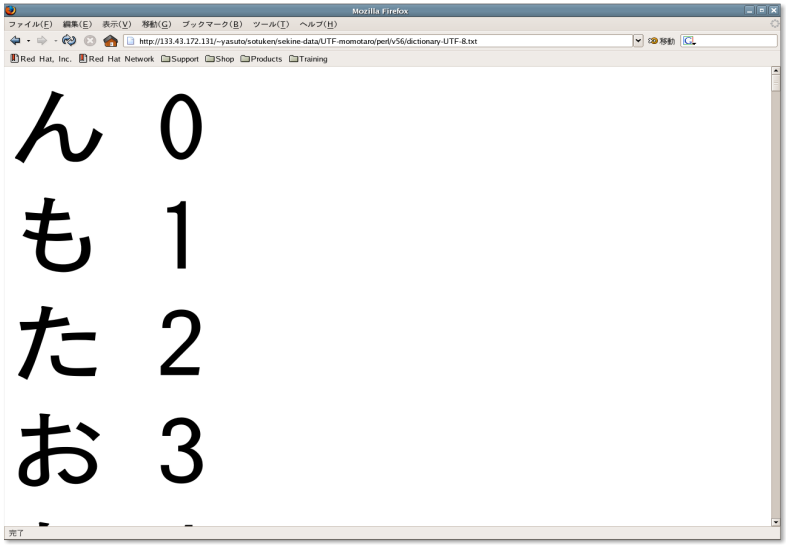
前処理1(pre1)

あいうえお順に番号付け

前処理2(pre2)

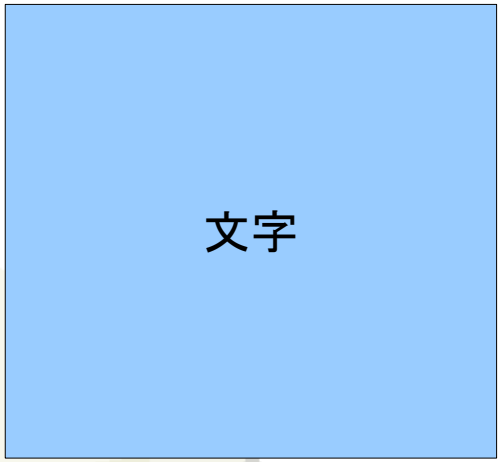
全てのデータの文字の出現数をカウントし、
出現頻度の多い順に番号付け

作成した1バイトのコードに変換

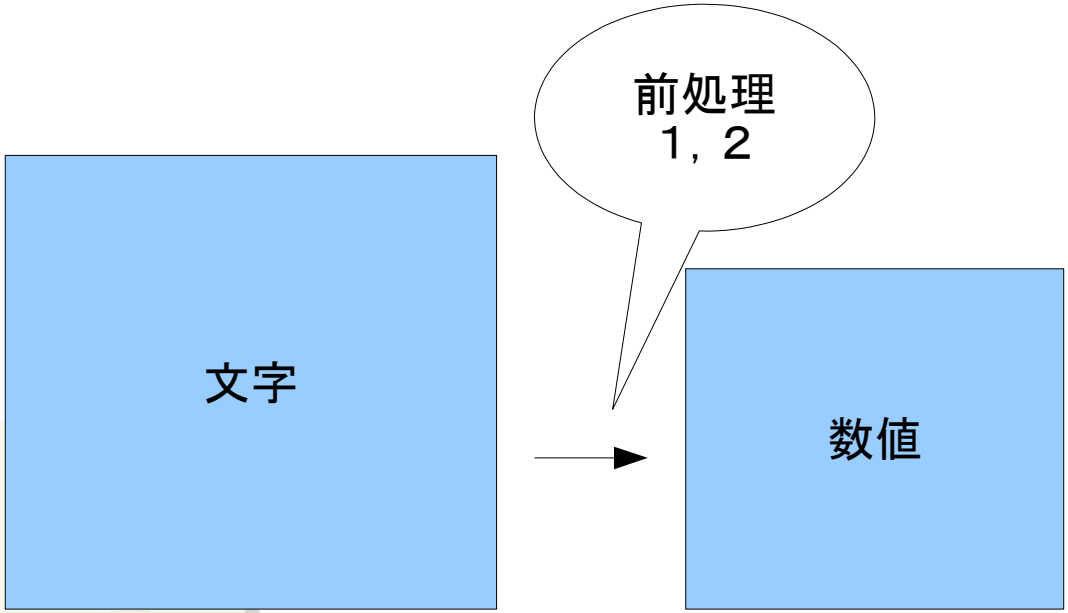


目次

前処理の流れ

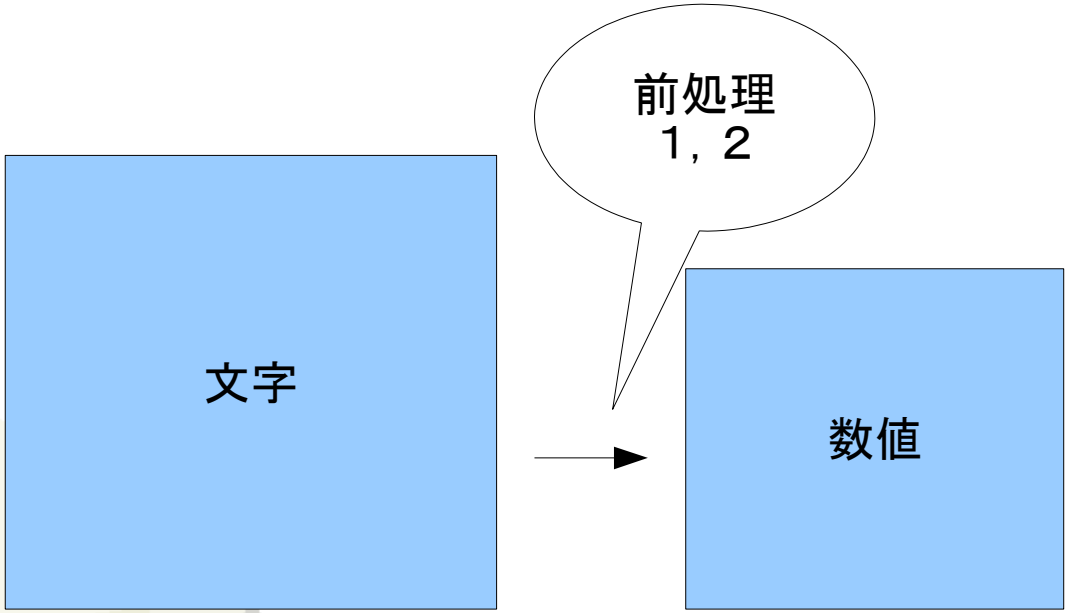


前処理の流れ



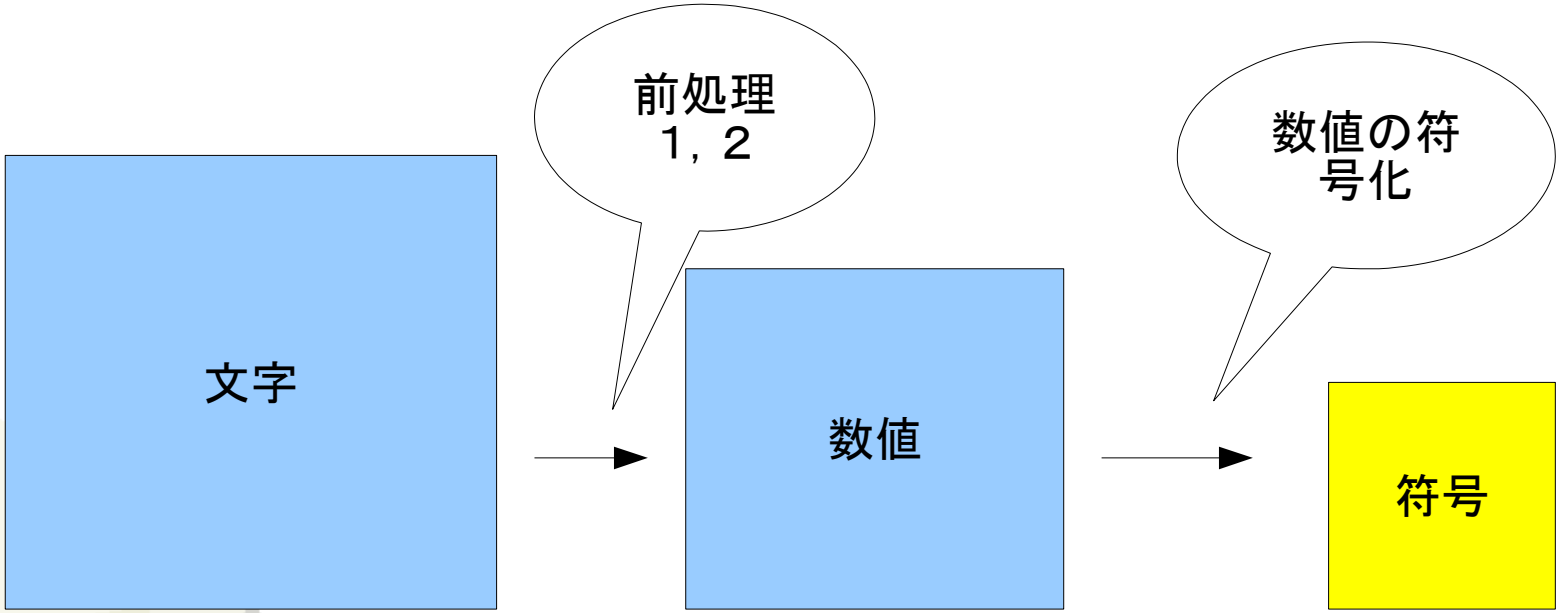
前処理の流れ

もっとファイルサイズを小さくしたい



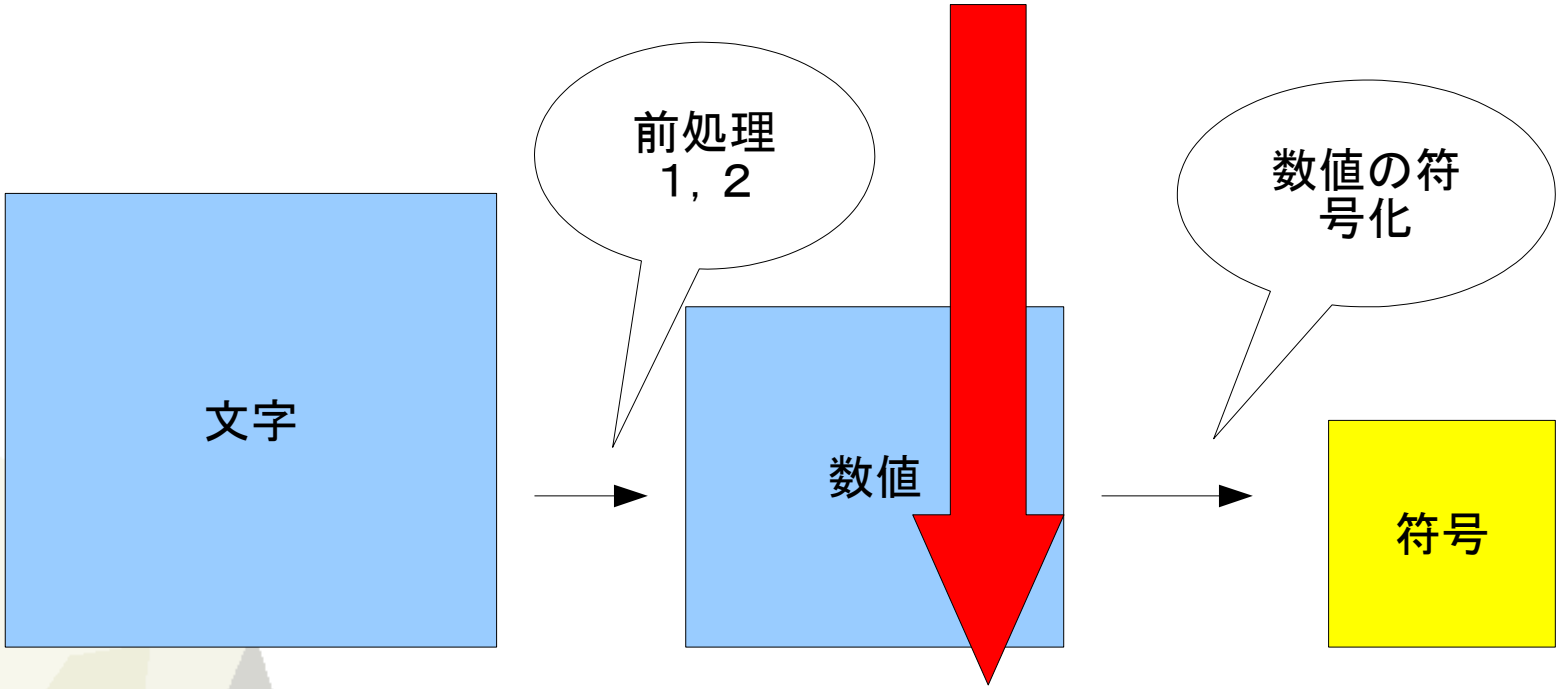
前処理の流れ

もっとファイルサイズを小さくしたい



前処理の流れ

もっとファイルサイズを小さくしたい



数値を符号化することによりファイルサイズを
小さくできるのでは

目次

目次

整数の符号化

整数の符号化とは、主にデータ圧縮において
整数を符号語にするための手法

整数をなるべく短い符号語で表現し、
全体の情報量を少なくする

本研究で使う符号

代表的な符号

アルファ符号

ガンマ符号

デルタ符号

ゴロム符号

ライス符号

本研究で使う符号

代表的な符号

アルファ符号

ガンマ符号

デルタ符号

ゴロム符号

ライス符号

ライス符号

ライス符号は実装が簡単なので、画像や音声の圧縮アルゴリズムの中で使われることがあります

目次

本研究での前処理

前処理1(pre1)

あいうえお順に番号付け
（1バイト）

前処理2(pre2)

全てのデータの文字の出現数をカウントし、
出現頻度の多い順に番号付け
（1バイト）

本研究での前処理

前処理1(pre1)

あいうえお順に番号付け
（1バイト）

前処理2(pre2)

全てのデータの文字の出現数をカウントし、
出現頻度の多い順に番号付け
（1バイト）

ライス符号化

前処理1-r(pre1-r)

あいうえお順に番号付け
（ライス符号化）

前処理2-r(pre2-r)

全てのデータの文字の出現数をカウントし、
出現頻度の多い順に番号付け
(ライス符号化)

本研究実験

圧縮(bzip2)前

圧縮(bzip2)後 先行研究

前処理1、2の
ファイルサイズ

前処理1、2の
ファイルサイズ

比較

本研究

前処理1-r、2-r(ライス符号化)
のファイルサイズ

前処理1-r、2-r(ライス符号化)
のファイルサイズ

本研究実験

圧縮(bzip2)前

圧縮(bzip2)後

先行研究

前処理1、2の
ファイルサイズ

前処理1、2の
ファイルサイズ

比較

本研究

前処理1-r、2-r(ライス符号化)
のファイルサイズ

前処理1-r、2-r(ライス符号化)
のファイルサイズ

ライス符号化が前処理で有効かを実験する

目次

ライス符号とは

ライス符号の説明の前にα符号について説明します

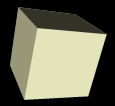
α符号とは

アルファ符号とは、一進法符号(単進符号, unary)とも呼ばれる、正の整数を表す可変長符号の一つ

n	α符号
0	1
1	01
2	001
3	0001
4	00001
5	000001
6	0000001
7	00000001
8	000000001
9	0000000001

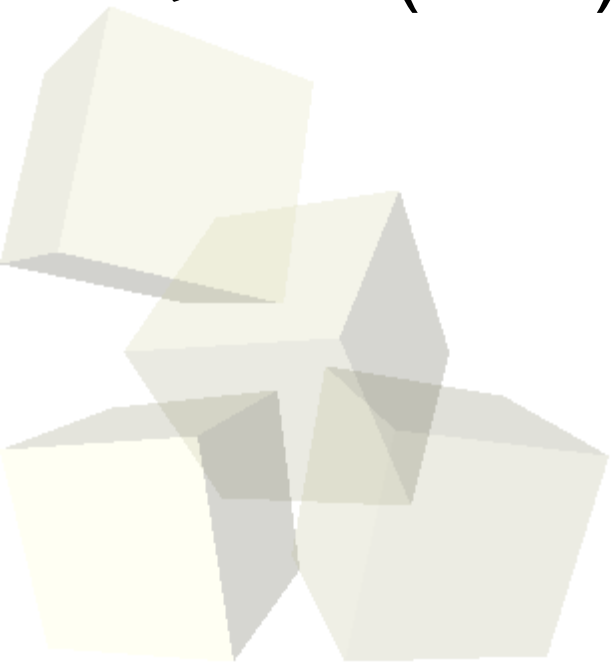
0 の個数で整数を表している

ライス符号の中で使われる



ライス符号とは

ライス (Rice) 符号はゴロム符号の特別な場合です



ライス符号とは

ライス符号はパラメータ b を使って
整数 n を符号化します

アルゴリズムは次のようになります

アルゴリズム

ゴロム符号

ライス符号

1. 商 p = n / b, 剰余 q = n % b を求める
2. p をα符号で符号化
3. $(\exists k \in N) b=2^k$場合、q の k ビットでバイナリ符号化

4. $!(\exists k \in N) b=2^k$の場合、q を CBT 符号で符号化

ライス符号の簡単な例(b=4)

b=4の場合で

n(0~15)をライス符号化してみる

ライス符号の簡単な例(b=4)

ライス符号

1. 商 p = n / b, 剰余 q = n % b を求める
2. p をα符号で符号化
3. ($\exists k \in N$)$b=2^k$場合、q のk ビットでバイナリ符号化

ライス符号の簡単な例(b=4)

ライス符号($b=4=2^2$)

1. 商 p = n / 4, 剰余 q = n % 4 を求める
2. p をα符号で符号化
3. ($\exists 2 \in N$)$b=2^2$場合、q の2ビットでバイナリ符号化

ライス符号の簡単な例(b=4)

ライス符号($b=4=2^2$)

1. 商 p = n / 4, 剰余 q = n % 4 を求める
2. p をα符号で符号化
3. ($\exists 2 \in N$)$b=2^2$場合、q の2ビットでバイナリ符号化

ライス符号の簡単な例(b=4)

n		p		q
0		0		0
1		0		1
2		0		2
3		0		3
4		1		0
5		1		1
6		1		2
7		1		3
8		2		0
9		2		1
10		2		2
11		2		3
12		3		0
13		3		1
14		3		2
15		3		3

ライス符号($b=4=2^2$)

1. 商 p = n / 4, 剰余 q = n % 4 を求める
2. p をα符号で符号化
3. ($\exists 2 \in N$)$b=2^2$場合、q の2ビットでバイナリ符号化

ライス符号の簡単な例(b=4)

n		p		q
0		0		0
1		0		1
2		0		2
3		0		3
4		1		0
5		1		1
6		1		2
7		1		3
8		2		0
9		2		1
10		2		2
11		2		3
12		3		0
13		3		1
14		3		2
15		3		3

ライス符号($b=4=2^2$)

1. 商 p = n / 4, 剰余 q = n % 4 を求める
2. p をα符号で符号化
3. ($\exists 2 \in N$)$b=2^2$場合、q の2ビットでバイナリ符号化

ライス符号の簡単な例(b=4)

n					p		q
0					1		0
1					1		1
2					1		2
3					1		3
4				0	1		0
5				0	1		1
6				0	1		2
7				0	1		3
8			0	0	1		0
9			0	0	1		1
10			0	0	1		2
11			0	0	1		3
12		0	0	0	1		0
13		0	0	0	1		1
14		0	0	0	1		2
15		0	0	0	1		3

ライス符号($b=4=2^2$)

1. 商 p = n / 4, 剰余 q = n % 4 を求める
2. p をα符号で符号化
3. ($\exists 2 \in N$)$b=2^2$場合、q の2ビットでバイナリ符号化

ライス符号の簡単な例(b=4)

n					p		q
0					1		0
1					1		1
2					1		2
3					1		3
4				0	1		0
5				0	1		1
6				0	1		2
7				0	1		3
8			0	0	1		0
9			0	0	1		1
10			0	0	1		2
11			0	0	1		3
12		0	0	0	1		0
13		0	0	0	1		1
14		0	0	0	1		2
15		0	0	0	1		3

ライス符号($b=4=2^2$)

1. 商 p = n / 4, 剰余 q = n % 4 を求める
2. p をα符号で符号化
3. ($\exists 2 \in N$)$b=2^2$場合、q の2ビットでバイナリ符号化

ライス符号の簡単な例(b=4)

n					p			q
0					1		0	0
1					1		0	1
2					1		1	0
3					1		1	1
4				0	1		0	0
5				0	1		0	1
6				0	1		1	0
7				0	1		1	1
8			0	0	1		0	0
9			0	0	1		0	1
10			0	0	1		1	0
11			0	0	1		1	1
12		0	0	0	1		0	0
13		0	0	0	1		0	1
14		0	0	0	1		1	0
15		0	0	0	1		1	1

ライス符号($b=4=2^2$)

1. 商 p = n / 4, 剰余 q = n % 4 を求める
2. p をα符号で符号化
3. ($\exists 2 \in N) b=2^2$場合、q の2ビットでバイナリ符号化

ライス符号の簡単な例(b=4)

n					p			q
0					1		0	0
1					1		0	1
2					1		1	0
3					1		1	1
4				0	1		0	0
5				0	1		0	1
6				0	1		1	0
7				0	1		1	1
8			0	0	1		0	0
9			0	0	1		0	1
10			0	0	1		1	0
11			0	0	1		1	1
12		0	0	0	1		0	0
13		0	0	0	1		0	1
14		0	0	0	1		1	0
15		0	0	0	1		1	1

ライス符号($b=4=2^2$)

1. 商 p = n / 4, 剰余 q = n % 4 を求める
2. p をα符号で符号化
3. ($\exists 2 \in N$)$b=2^2$場合、q の2ビットでバイナリ符号化

次にb=8の場合と比較してみます

ライス符号の簡単な例(b=4,8)

b=4

n					p			q		
0					1		0	0		
1					1		0	1		
2					1		1	0		
3					1		1	1		
4				0	1		0	0		
5				0	1		0	1		
6				0	1		1	0		
7				0	1		1	1		
8			0	0	1		0	0		
9			0	0	1		0	1		
10			0	0	1		1	0		
11			0	0	1		1	1		
12		0	0	0	1		0	0		
13		0	0	0	1		0	1		
14		0	0	0	1		1	0		
15		0	0	0	1		1	1		

b=8

n				p				q		
0				1			0	0	0	
1				1			0	0	1	
2				1			0	1	0	
3				1			0	1	1	
4				1			1	0	0	
5				1			1	0	1	
6				1			1	1	0	
7		0		1			1	1	1	
8		0		1			0	0	0	
9		0		1			0	0	1	
10		0		1			0	1	0	
11		0		1			0	1	1	
12		0		1			1	0	0	
13		0		1			1	0	1	
14		0		1			1	1	0	
15		0		1			1	1	1	

ライス符号の簡単な例(b=4,8)

b=4

n					p			q			n			p				q
0					1		0	0			0			1		0	0	0
1					1		0	1			1			1		0	0	1
2					1		1	0			2			1		0	1	0
3					1		1	1			3			1		0	1	1
4				0	1		0	0			4			1		1	0	0
5				0	1		0	1			5			1		1	0	1
6				0	1		1	0			6			1		1	1	0
7				0	1		1	1			7		0	1		1	1	1
8			0	0	1		0	0			8		0	1		0	0	0
9			0	0	1		0	1			9		0	1		0	0	1
10																		
11																		
12																		
13																		
14																		
15		0	0	0	1		1	1			15		0	1		1	1	1

b=8

このように、ライス符号はパラメータ b を変更すると符号語も変化します

パラメータbを変更した際のライス符号の変化

今度はb=4、8、16、32の場合で

n(0~80)をライス符号化してみた

前処理1、2
で使われる数値の
範囲

パラメータbを変更した際のライス符号の変化

b=4

```
46 [45]:000000000000101↓
47 [46]:000000000000110↓
48 [47]:000000000000111↓
49 [48]:0000000000000100↓
50 [49]:0000000000000101↓
51 [50]:0000000000000110↓
52 [51]:0000000000000111↓
53 [52]:00000000000000100↓
54 [53]:00000000000000101↓
55 [54]:00000000000000110↓
56 [55]:00000000000000111↓
57 [56]:000000000000000100↓
58 [57]:000000000000000101↓
59 [58]:000000000000000110↓
60 [59]:000000000000000111↓
61 [60]:0000000000000000100↓
62 [61]:0000000000000000101↓
63 [62]:0000000000000000110↓
64 [63]:0000000000000000111↓
65 [64]:00000000000000000100↓
66 [65]:00000000000000000101↓
67 [66]:00000000000000000110↓
68 [67]:00000000000000000111↓
69 [68]:000000000000000000100↓
70 [69]:000000000000000000101↓
71 [70]:000000000000000000110↓
72 [71]:000000000000000000111↓
73 [72]:0000000000000000000100↓
74 [73]:0000000000000000000101↓
75 [74]:0000000000000000000110↓
76 [75]:0000000000000000000111↓
77 [76]:00000000000000000000100↓
78 [77]:00000000000000000000101↓
79 [78]:00000000000000000000110↓
80 [79]:00000000000000000000111↓
81 [EOF]
```

b=8

```
46 [45]:00000001101↓
47 [46]:00000001110↓
48 [47]:00000001111↓
49 [48]:00000001000↓
50 [49]:00000001001↓
51 [50]:00000001010↓
52 [51]:00000001011↓
53 [52]:00000001100↓
54 [53]:00000001101↓
55 [54]:00000001110↓
56 [55]:00000001111↓
57 [56]:000000001000↓
58 [57]:000000001001↓
59 [58]:000000001010↓
60 [59]:000000001011↓
61 [60]:000000001100↓
62 [61]:000000001101↓
63 [62]:000000001110↓
64 [63]:000000001111↓
65 [64]:0000000001000↓
66 [65]:0000000001001↓
67 [66]:0000000001010↓
68 [67]:0000000001011↓
69 [68]:0000000001100↓
70 [69]:0000000001101↓
71 [70]:0000000001110↓
72 [71]:0000000001111↓
73 [72]:00000000001000↓
74 [73]:00000000001001↓
75 [74]:00000000001010↓
76 [75]:00000000001011↓
77 [76]:00000000001100↓
78 [77]:00000000001101↓
79 [78]:00000000001110↓
80 [79]:00000000001111↓
81 [EOF]
```

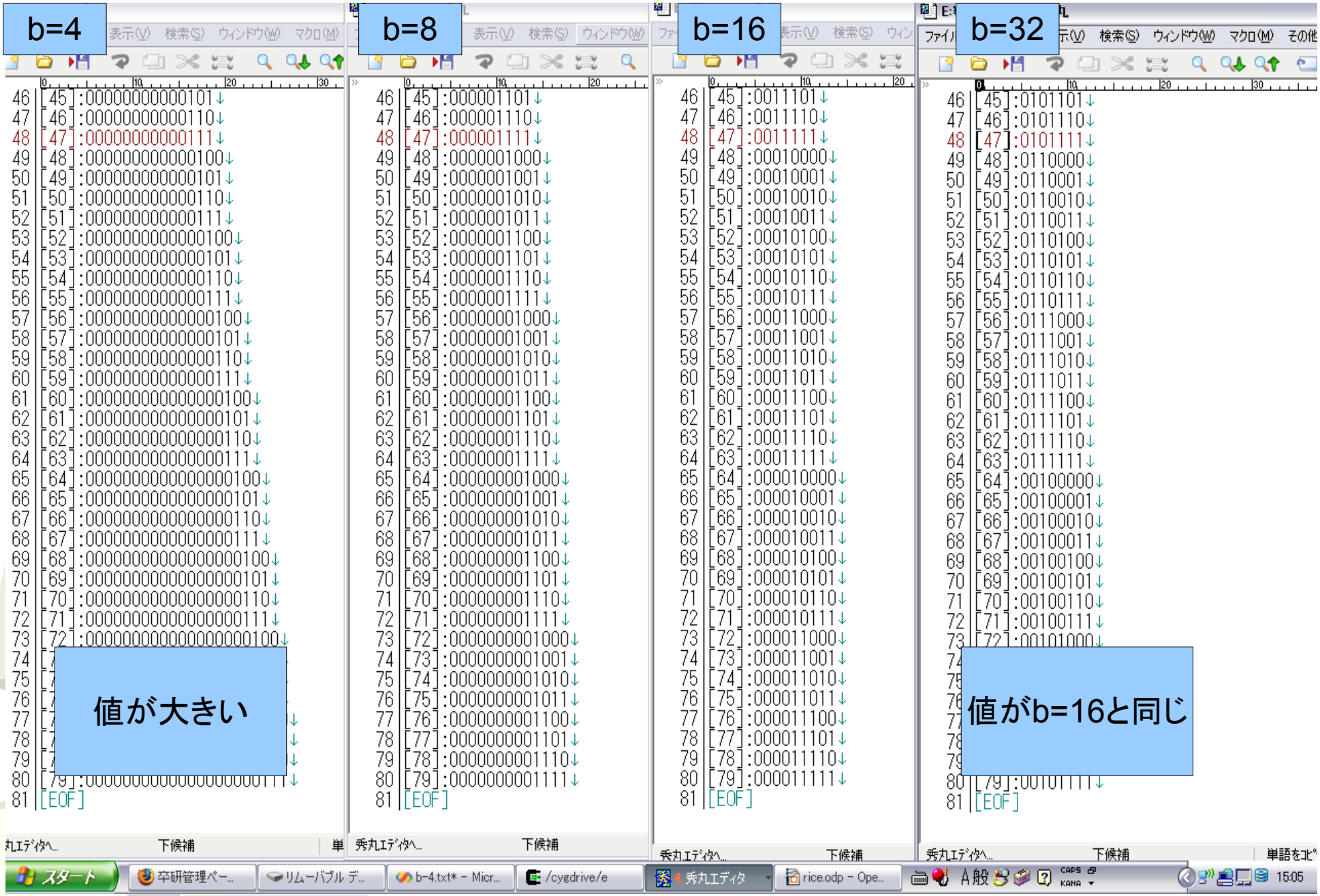
b=16

```
46 [45]:0011101↓
47 [46]:0011110↓
48 [47]:0011111↓
49 [48]:00010000↓
50 [49]:00010001↓
51 [50]:00010010↓
52 [51]:00010011↓
53 [52]:00010100↓
54 [53]:00010101↓
55 [54]:00010110↓
56 [55]:00010111↓
57 [56]:00011000↓
58 [57]:00011001↓
59 [58]:00011010↓
60 [59]:00011011↓
61 [60]:00011100↓
62 [61]:00011101↓
63 [62]:00011110↓
64 [63]:00011111↓
65 [64]:000010000↓
66 [65]:000010001↓
67 [66]:000010010↓
68 [67]:000010011↓
69 [68]:000010100↓
70 [69]:000010101↓
71 [70]:000010110↓
72 [71]:000010111↓
73 [72]:000011000↓
74 [73]:000011001↓
75 [74]:000011010↓
76 [75]:000011011↓
77 [76]:000011100↓
78 [77]:000011101↓
79 [78]:000011110↓
80 [79]:000011111↓
81 [EOF]
```

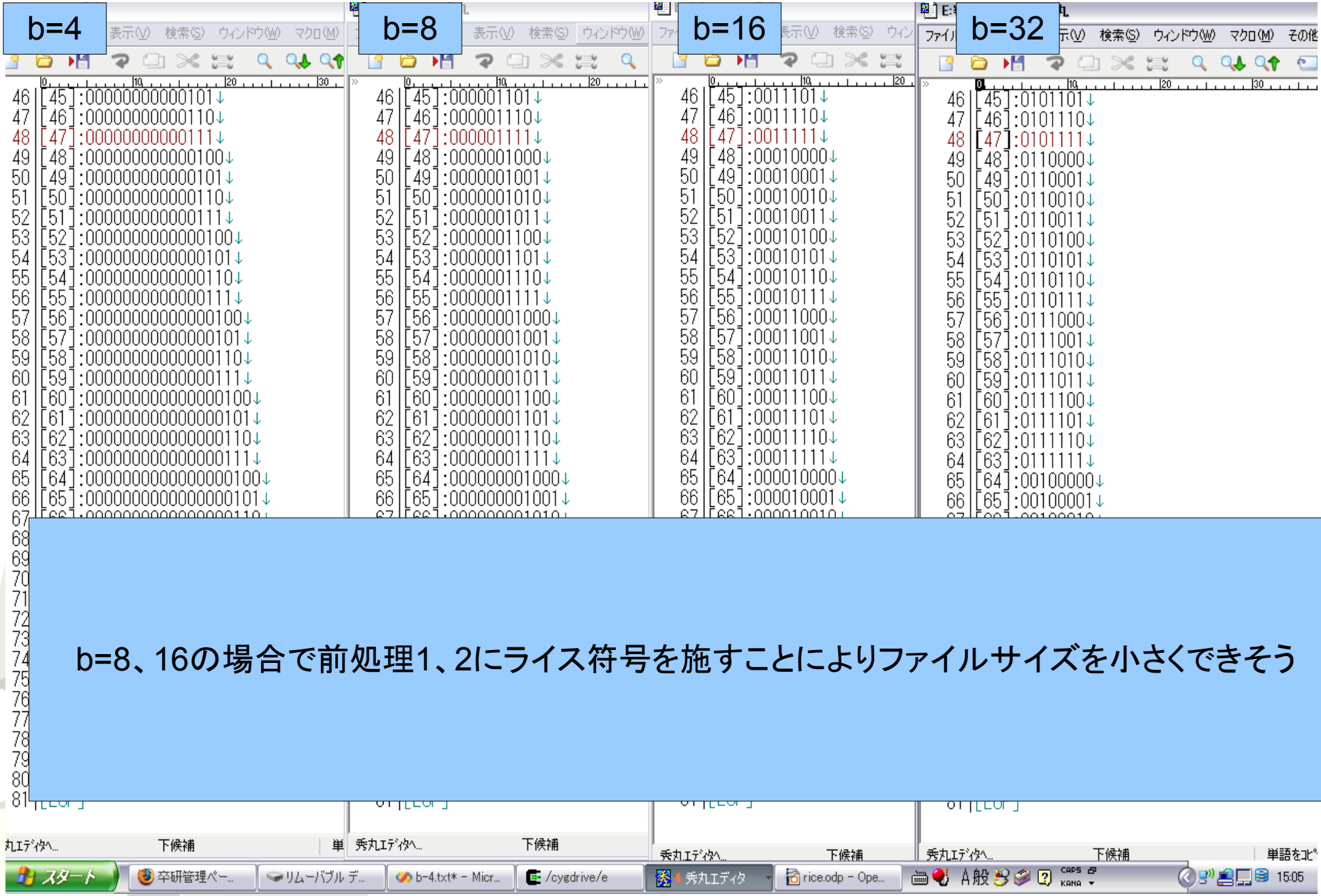
b=32

```
46 [45]:01011101↓
47 [46]:01011110↓
48 [47]:01011111↓
49 [48]:01100000↓
50 [49]:01100001↓
51 [50]:01100010↓
52 [51]:01100011↓
53 [52]:01100100↓
54 [53]:01100101↓
55 [54]:01100110↓
56 [55]:01100111↓
57 [56]:01101000↓
58 [57]:01101001↓
59 [58]:01101010↓
60 [59]:01101011↓
61 [60]:01101100↓
62 [61]:01101101↓
63 [62]:01101110↓
64 [63]:01101111↓
65 [64]:00100000↓
66 [65]:00100001↓
67 [66]:00100010↓
68 [67]:00100011↓
69 [68]:00100100↓
70 [69]:00100101↓
71 [70]:00100110↓
72 [71]:00100111↓
73 [72]:00101000↓
74 [73]:00101001↓
75 [74]:00101010↓
76 [75]:00101011↓
77 [76]:00101100↓
78 [77]:00101101↓
79 [78]:00101110↓
80 [79]:00101111↓
81 [EOF]
```

パラメータbを変更した際のライス符号の変化



パラメータbを変更した際のライス符号の変化



本研究での前処理

前処理1(pre1)

あいうえお順に番号付け
（1バイト）

前処理2(pre2)

全てのデータの文字の出現数をカウントし、
出現頻度の多い順に番号付け
（1バイト）

ライス符号化

pre1-r	pre1-r	pre2-r	pre2-r
b=8	b=16	b=8	b=16

ライス符号化が前処理で有効か

目次

圧縮前のファイルサイズ比較(ライス符号(b=8))

前処理1(pre1)

あいうえお順に番号付け
(1バイト)

前処理2(pre2)

全てのデータの文字の出現数をカウントし、
出現頻度の多い順に番号付け
(1バイト)

比較

前処理1-r(pre1-r)

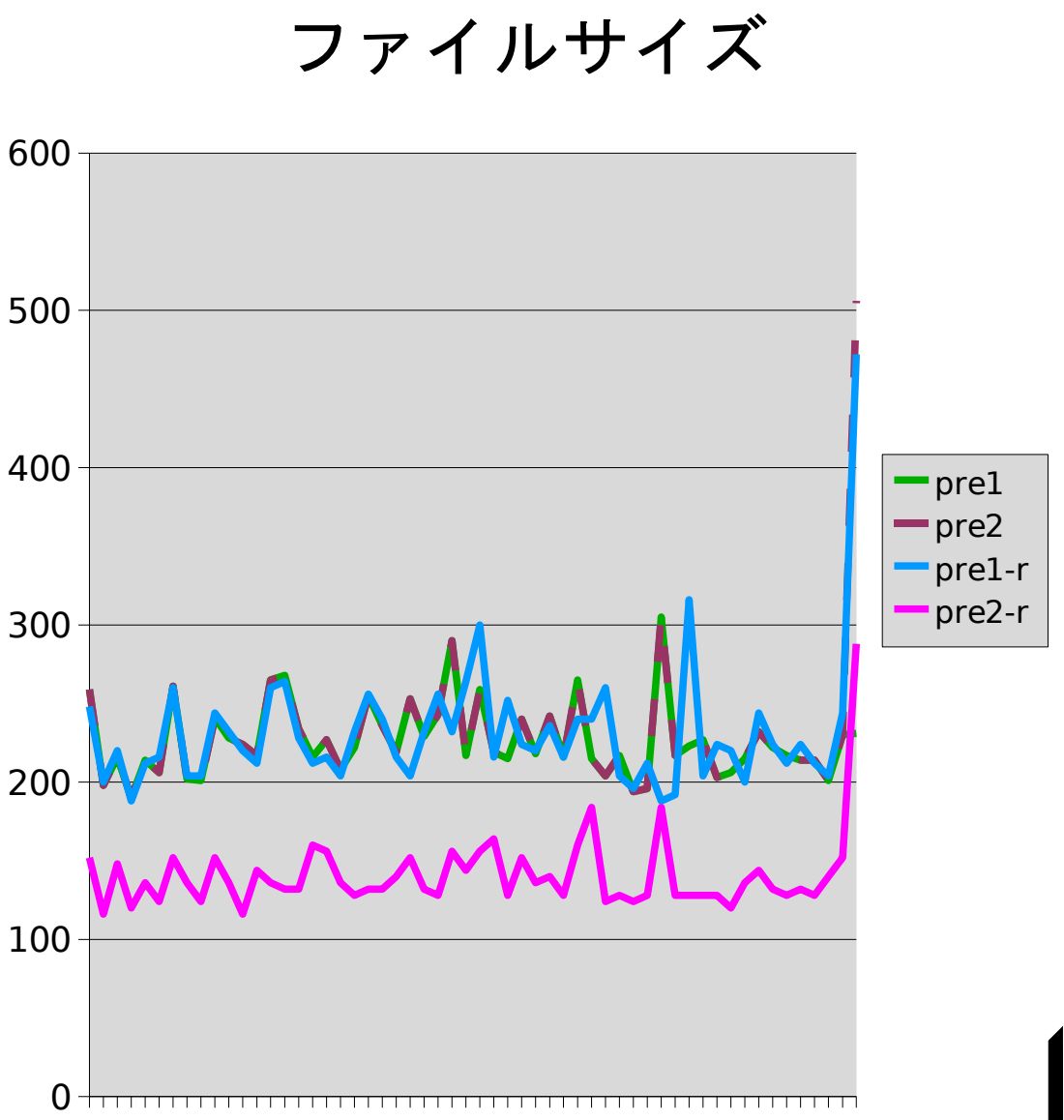
b=8
(ライス符号化)

前処理2-r(pre2-r)

b=8
(ライス符号化)

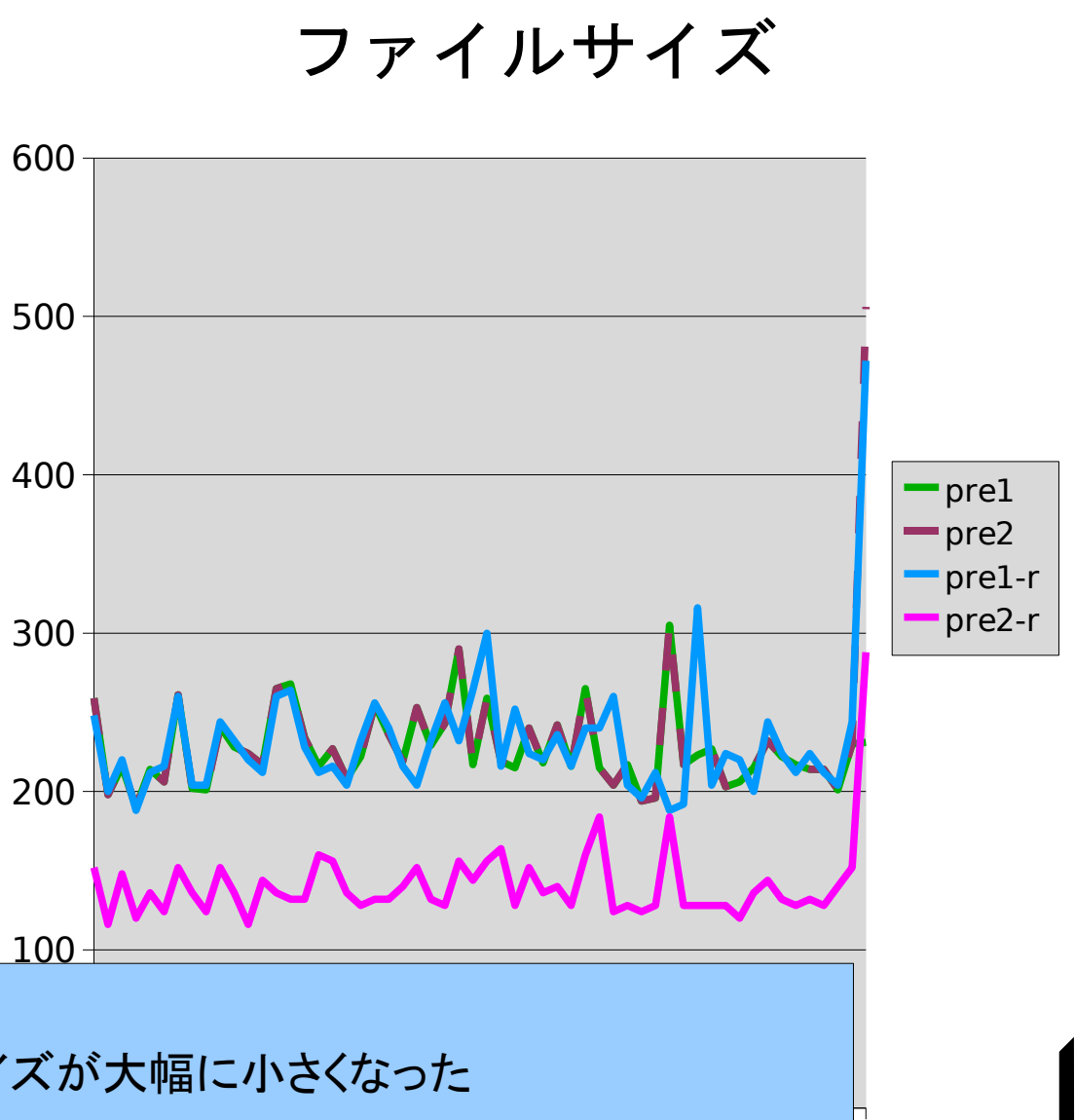
圧縮前のファイルサイズ比較(ライス符号(b=8))

Filesize	pre1	pre2	pre1-r	pre2-r
aichi	259	259	248	152
akita	198	198	200	116
aomori	216	216	220	148
chiba1	190	190	188	120
chiba2	214	214	212	136
ehime	206	206	216	124
fukuoka	261	261	260	152
fukushima	202	202	204	136
gifu	201	201	204	116
gunma	241	241	244	144
hiroshima	228	228	232	136
hokkaido1	224	224	220	132
hokkaido2	217	217	212	132
hyogo1	265	265	260	160
hyogo2	268	268	264	156
ibaraki	234	234	228	136
ishikawa	216	216	212	128



圧縮前のファイルサイズ比較(ライス符号(b=8))

Filesize	pre1	pre2	pre1-r	pre2-r
aichi	259	259	248	152
akita	198	198	200	116
aomori	216	216	220	148
chiba1	190	190	188	120
chiba2	214	214	212	136
ehime	206	206	216	124
fukuoka	261	261	260	152
fukushima	202	202	204	136
gifu	201	201	204	116
gunma	241	241	244	144
hiroshima	228	228	232	136
hokkaido1	224	224	220	132
hokkaido2	217	217	212	132
hyogo1	265	265	260	160



pre2-rにおいてファイルサイズが大幅に小さくなった

圧縮前のファイルサイズ比較(ライス符号(b=16))

前処理1(pre1)

あいうえお順に番号付け
(1バイト)

前処理2(pre2)

全てのデータの文字の出現数をカウントし、
出現頻度の多い順に番号付け
(1バイト)

比較

前処理1-r(pre1-r)

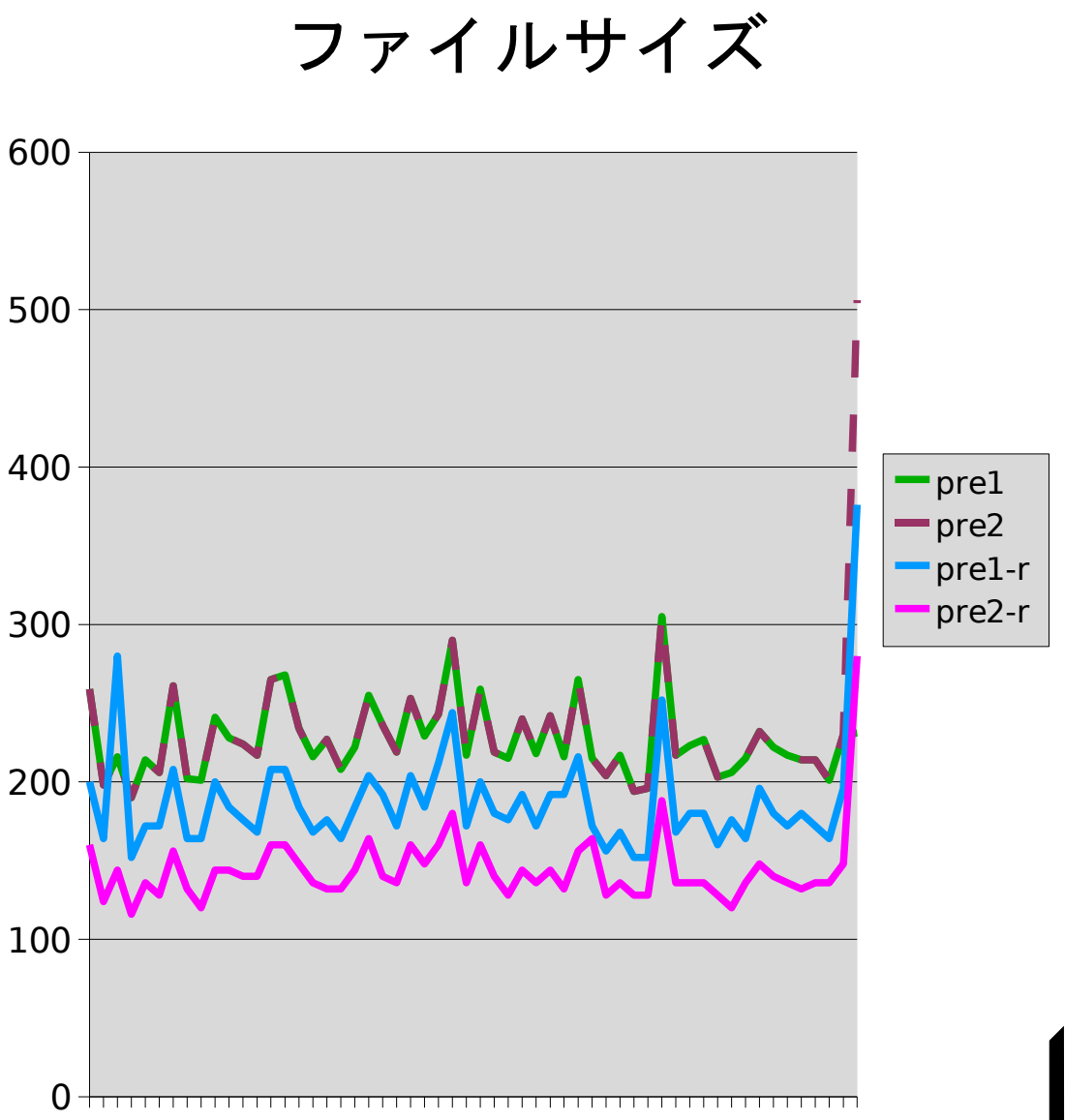
b=16
(ライス符号化)

前処理2-r(pre2-r)

b=16
(ライス符号化)

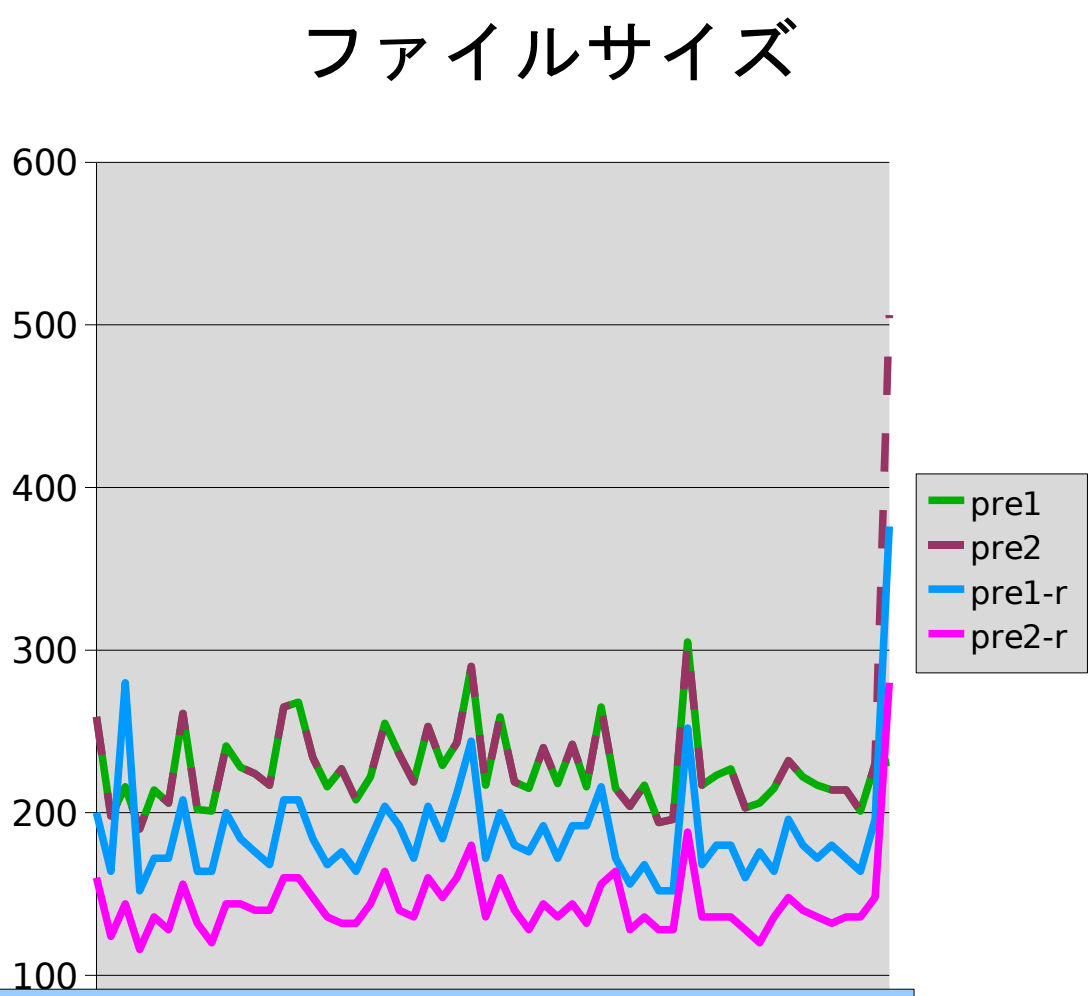
圧縮前のファイルサイズ比較(ライス符号(b=16))

Filesize	pre1	pre2	pre1-r	pre2-r
aichi	259	259	200	160
akita	198	198	164	124
aomori	216	216	180	144
chiba1	190	190	152	116
chiba2	214	214	172	136
ehime	206	206	172	128
fukuoka	261	261	208	156
fukushima	202	202	164	132
gifu	201	201	164	120
gunma	241	241	200	144
hiroshima	228	228	184	144
hokkaido1	224	224	176	140
hokkaido2	217	217	168	140
hyogo1	265	265	208	160
hyogo2	268	268	208	160
ibaraki	234	234	184	148
ishikawa	216	216	168	136



圧縮前のファイルサイズ比較(ライス符号(b=16))

Filesize	pre1	pre2	pre1-r	pre2-r
aichi	259	259	200	160
akita	198	198	164	124
aomori	216	216	180	144
chiba1	190	190	152	116
chiba2	214	214	172	136
ehime	206	206	172	128
fukuoka	261	261	208	156
fukushima	202	202	164	132
gifu	201	201	164	120
gunma	241	241	200	144
hiroshima	228	228	184	144
hokkaido1	224	224	176	140
hokkaido2	217	217	168	140
hyogo1	265	265	208	160



pre1-r、2-rにおいて大幅にファイルサイズが小さくなった

圧縮後のファイルサイズ比較(ライス符号(b=8))

前処理1(pre1)

あいうえお順に番号付け
(1バイト)

bzip2

前処理2(pre2)

全てのデータの文字の出現数
をカウントし、
出現頻度の多い順に番号付け
(1バイト)

bzip2

比較

前処理1-r(pre1-r)

b=8
(ライス符号化)

bzip2

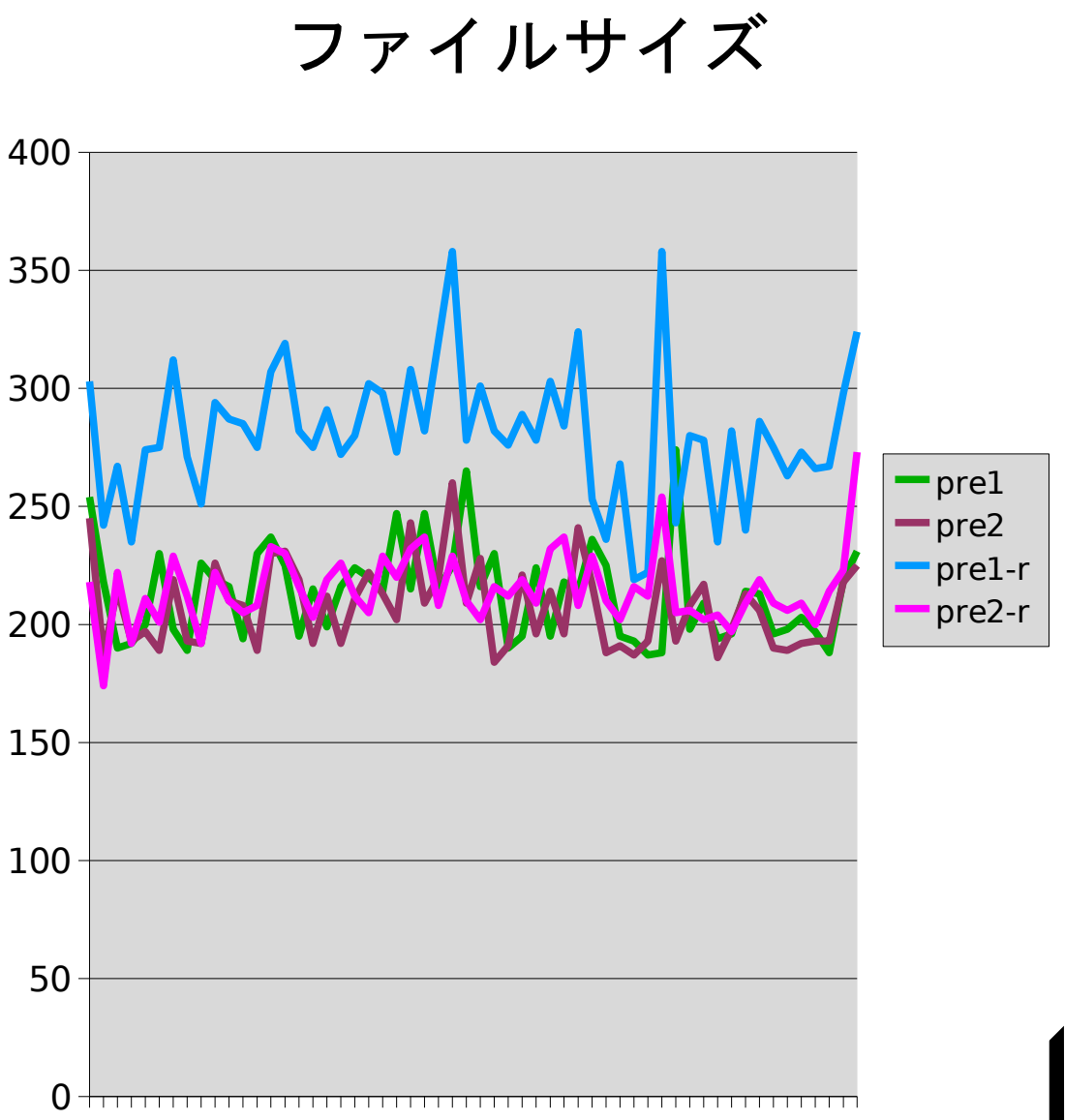
前処理2-r(pre2-r)

b=8
(ライス符号化)

bzip2

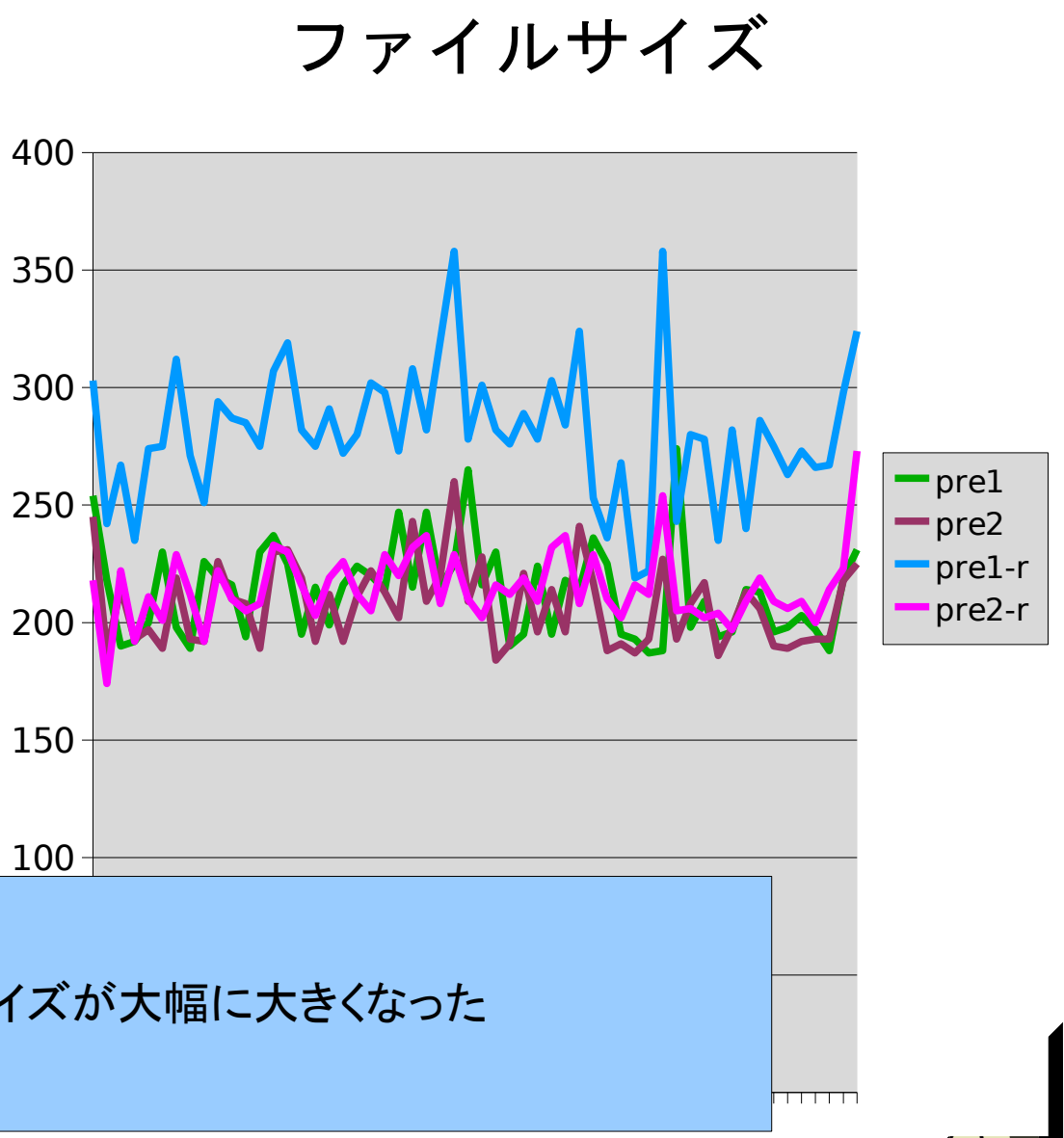
圧縮後のファイルサイズ比較(ライス符号(b=8))

Filesize	pre1	pre2	pre1-r	pre2-r
aichi	254	245	303	218
akita	218	187	242	174
aomori	190	215	267	222
chiba1	192	193	235	192
chiba2	200	197	274	211
ehime	230	189	275	201
fukuoka	198	219	312	229
fukushima	189	193	271	212
gifu	226	192	251	192
gunma	219	226	294	222
hiroshima	216	210	287	210
hokkaido1	194	208	285	205
hokkaido2	230	189	275	208
hyogo1	237	230	307	233
hyogo2	225	231	319	230
ibaraki	195	219	282	216
ishikawa	215	192	275	203



圧縮後のファイルサイズ比較(ライス符号(b=8))

Filesize	pre1	pre2	pre1-r	pre2-r
aichi	254	245	303	218
akita	218	187	242	174
aomori	190	215	267	222
chiba1	192	193	235	192
chiba2	200	197	274	211
ehime	230	189	275	201
fukuoka	198	219	312	229
fukushima	189	193	271	212
gifu	226	192	251	192
gunma	219	226	294	222
hiroshima	216	210	287	210
hokkaido1	194	208	285	205
hokkaido2	230	189	275	208
hyogo				
hyogo				
ibaraki				
ishikawa	215	192	275	205



pre1-rにおいてファイルサイズが大幅に大きくなった

圧縮後のファイルサイズ比較(ライス符号(b=16))

前処理1(pre1)

あいうえお順に番号付け
(1バイト)

bzip2

前処理2(pre2)

全てのデータの文字の出現数をカウントし、
出現頻度の多い順に番号付け
(1バイト)

bzip2

比較

前処理1-r(pre1-r)

b=16
(ライス符号化)

bzip2

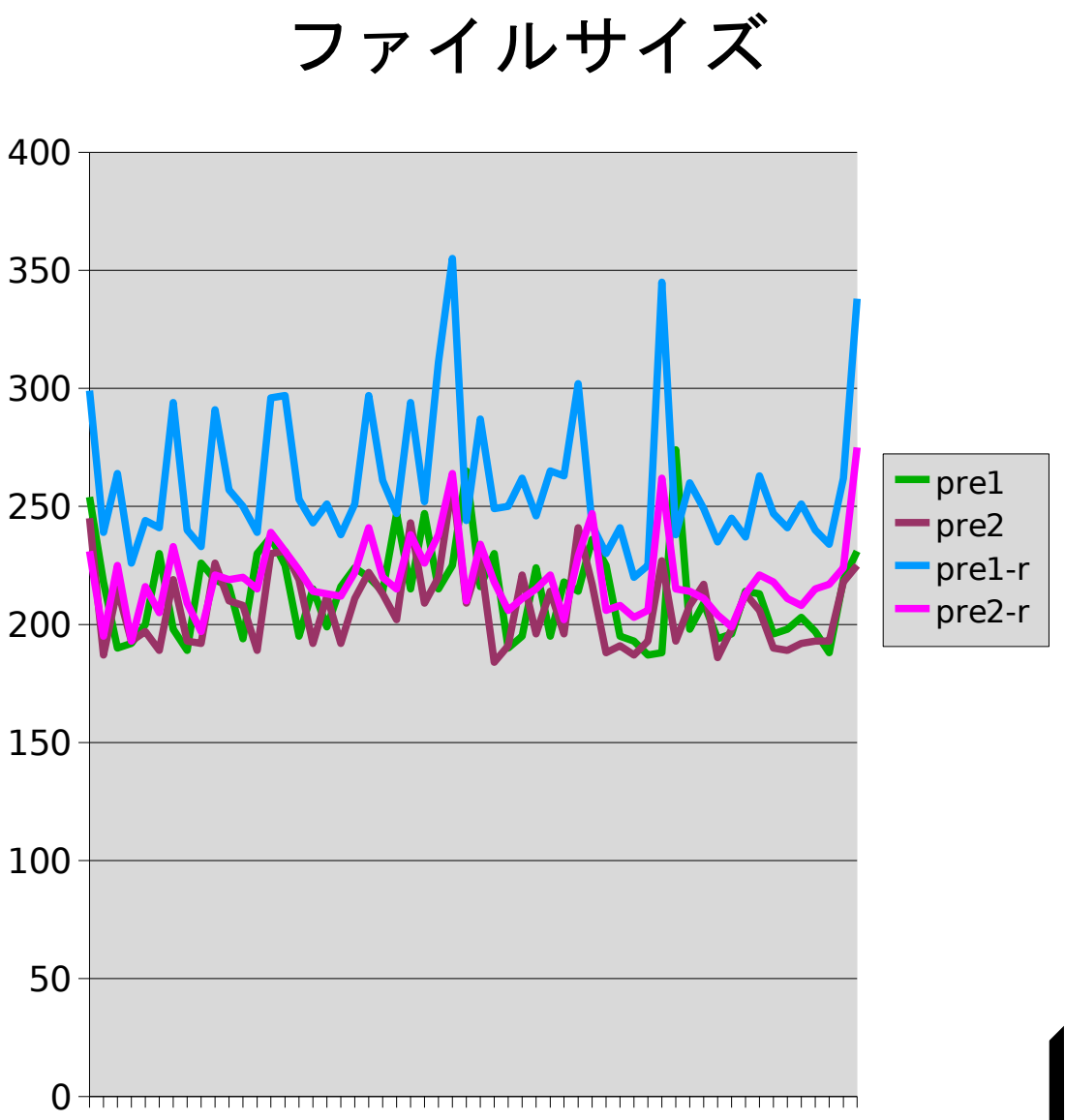
前処理2-r(pre2-r)

b=16
(ライス符号化)

bzip2

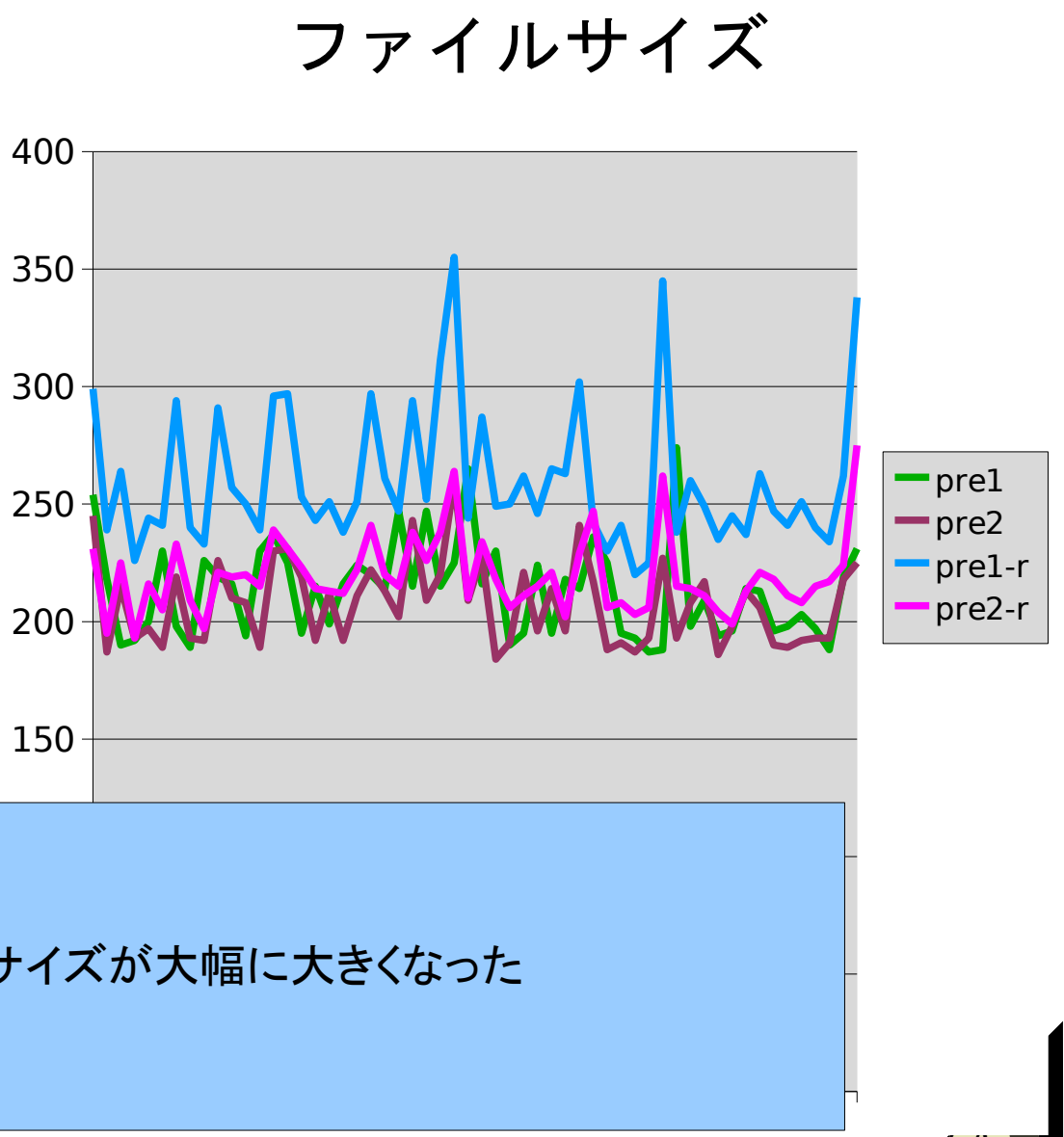
圧縮後のファイルサイズ比較(ライス符号(b=16))

Filesize	pre1	pre2	pre1-r	pre2-r
aichi	254	245	299	231
akita	218	187	239	195
aomori	190	215	264	225
chiba1	192	193	226	193
chiba2	200	197	244	216
ehime	230	189	241	205
fukuoka	198	219	294	233
fukushima	189	193	240	209
gifu	226	192	233	197
gunma	219	226	291	221
hiroshima	216	210	257	219
hokkaido1	194	208	250	220
hokkaido2	230	189	239	215
hyogo1	237	230	296	239
hyogo2	225	231	297	231
ibaraki	195	219	261	223
ishikawa	215	192	247	214



圧縮後のファイルサイズ比較(ライス符号(b=16))

Filesize	pre1	pre2	pre1-r	pre2-r
aichi	254	245	299	231
akita	218	187	239	195
aomori	190	215	264	225
chiba1	192	193	226	193
chiba2	200	197	244	216
ehime	230	189	241	205
fukuoka	198	219	294	233
fukushima	189	193	240	209
gifu	226	192	233	197
gunma	219	226	291	221
hiroshima	216	210	257	219
hokkaido1	194	208	250	220
hokk				
hyog				
hyog				
ibara				
ishikawa	215	192	247	214



pre1-rにおいてファイルサイズが大幅に大きくなった

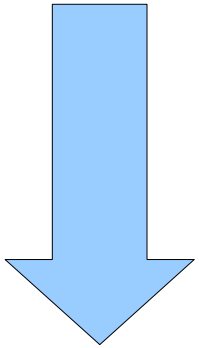
目次

考察

ライス符号化を施すことにより
圧縮前のファイルサイズは小さくなったが
圧縮後のファイルサイズは大きくなった

考察

ライス符号化を施すことにより
圧縮前のファイルサイズは小さくなったが
圧縮後のファイルサイズは大きくなった



前処理にライス符号化を施すことが有効かどうか
は現時点ではわからなかった

今後の課題

bzip2以外の圧縮方法で実験し、ライス符号化が方言の自動分類における精度の向上に有効か確かめる

おわり