Encoding and modeling for set compression

N. Jesper Larsson (`jesl`)
IT University of Copenhagen (`itu.dk`)

NII Shonan Meeting, 27–30 Sep. 2013

Metadata about this work

- Started as broad attack on *set compression*
- Contribute one algorithm + smaller points
- Not particularly about data structures or time complexity, but setting goals/ideals and finding potentials

Events, sequences, sets

Event (“circumstance”) sequence $E_1, \ldots, E_n$

$$\Pr(E_i | E_1 \cap \cdots \cap E_{i-1})$$

$$\sum_i -\log_2 \Pr(E_i | E_1 \cap \cdots \cap E_{i-1}) \text{ bits}$$

Encoder decoder share premises for what E_i mean in terms of specifying message (data)

- Seqence: a certain character is at position *i*
- Set: a certain element is in the set (note: no *i*)
- Set, alt.: a certain number of elements have a certain property
- …

Fields of application/ previous work

- Component (e.g. ψ)
- Inverted index
- Dictionary
- Data mining (measuring ratio), web graphs, …

Universe?

Encoding a set (or many sets) S,
elements drawn from universe U

$$S \in U$$

$$|S| < |U|$$

Universe?

$$|S| \sim \frac{1}{c}|U|$$

small constant

$$|S| \ll |U| < \infty$$

- fixed-length strings?
- characters? patterns?
- probability distribution? (Reznik 2011, Varshney & Goyal)

$$|U| = \infty$$

e.g. re-pair dictionary

Narrow focus, for now:

- U may be much larger than S
- Dependencies between elements
- Elements:
    - Integers $\in [0, |U|)$
    - = bitstrings of length $\lceil \log_2 |U| \rceil$

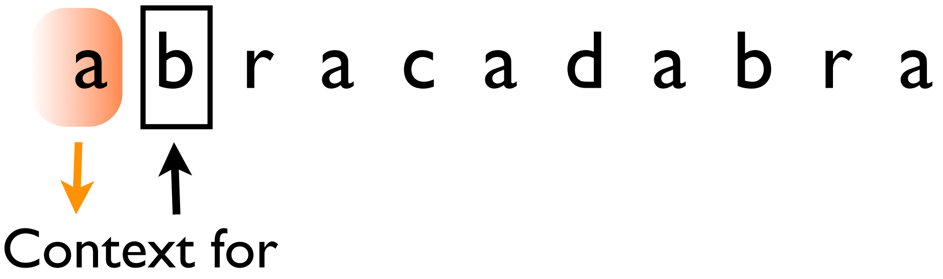
Solved?

- Set: { 4, 9, 11, 14, 16, 17, 20, 21 }
- Gaps: { 4, 4, 1, 2, 1, 0, 2, 0 }
- Geometric distribution
  $\Pr(\text{gap size k}) = (1-p)^{k-1}p, \;\; p = \Pr(x \in S) = |S|/|U|$
  Optimal code: Golomb (or arithmetic)

Known (?) method 2: yes/no code

- Arithmetic code for binary source:
  for each element of U, encode whether in S
- Estimate $p_x = \Pr(x \in S)$
- Use probability ranges $[0, p_x), [p_x, 1)$

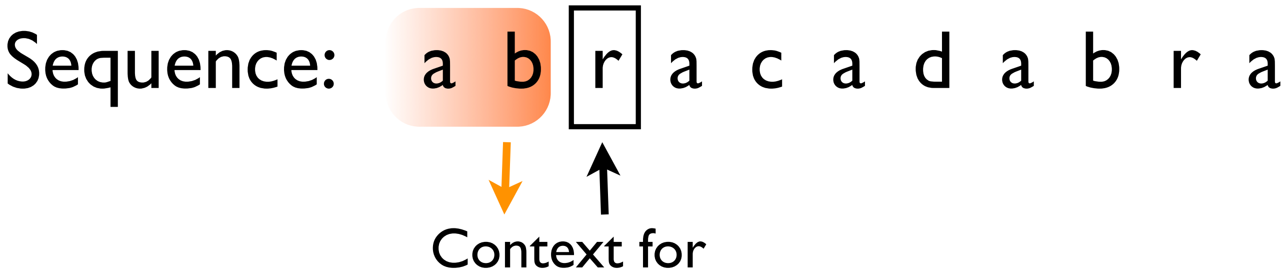
Context?



Sequence: a b r a c a d a b r a

Context for

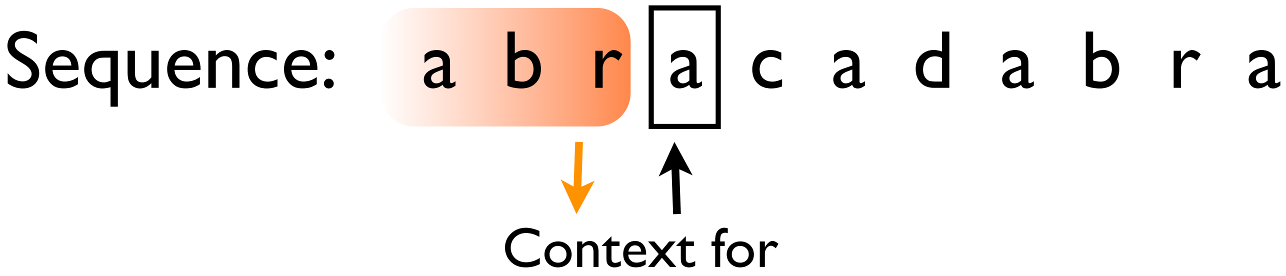
Context?



Sequence: a b r a c a d a b r a

Context for

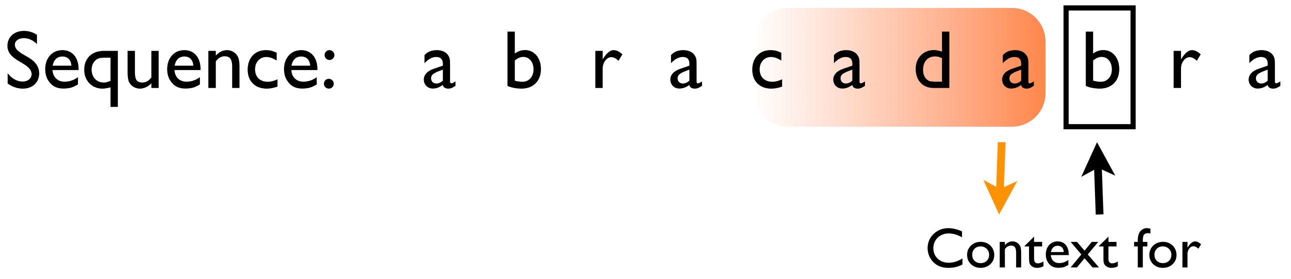
Context?



Sequence: a b r a c a d a b r a

Context for

Context?

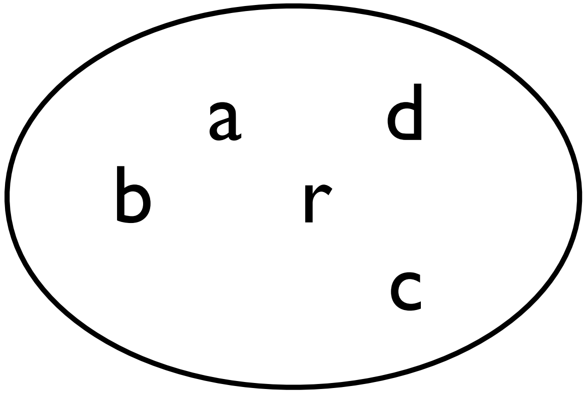


Sequence: a b r a c a d a b r a

Context for

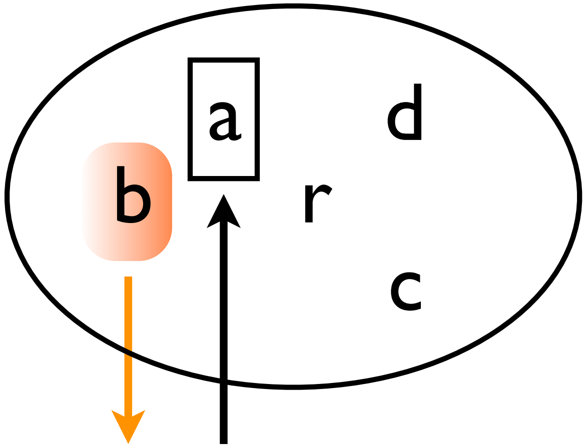
Context?

Set:



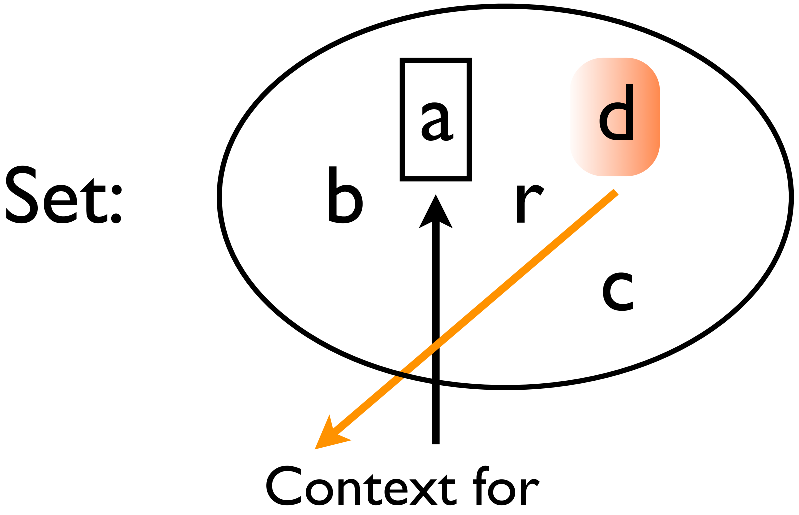
Context?

Set:

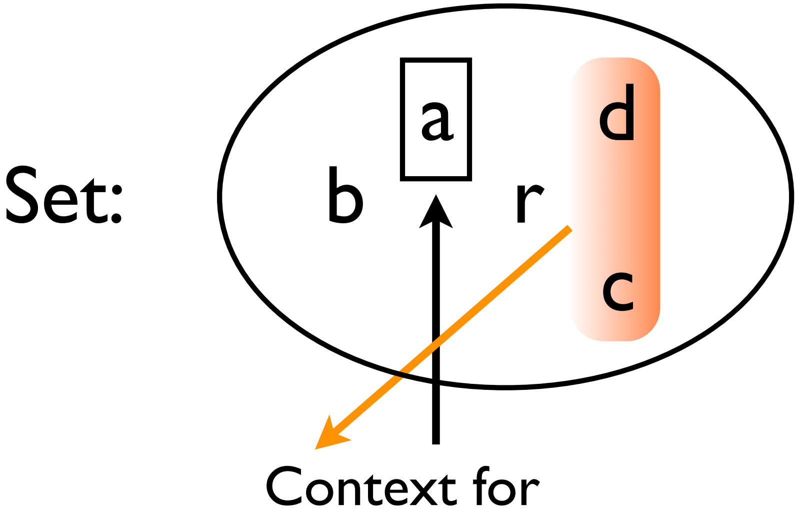


Context for

Context?

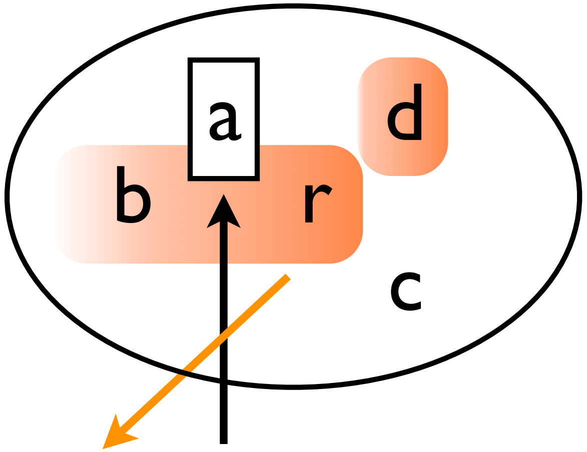


Context?



Context?

Set:



Context for

Hinting one path: context partitioning

- Partition into subsets for dependencies:
- … strong between subsets
- … weak between elements in same subset
- Condition probabilities on subsets encoded
- Order of subset transmission not important

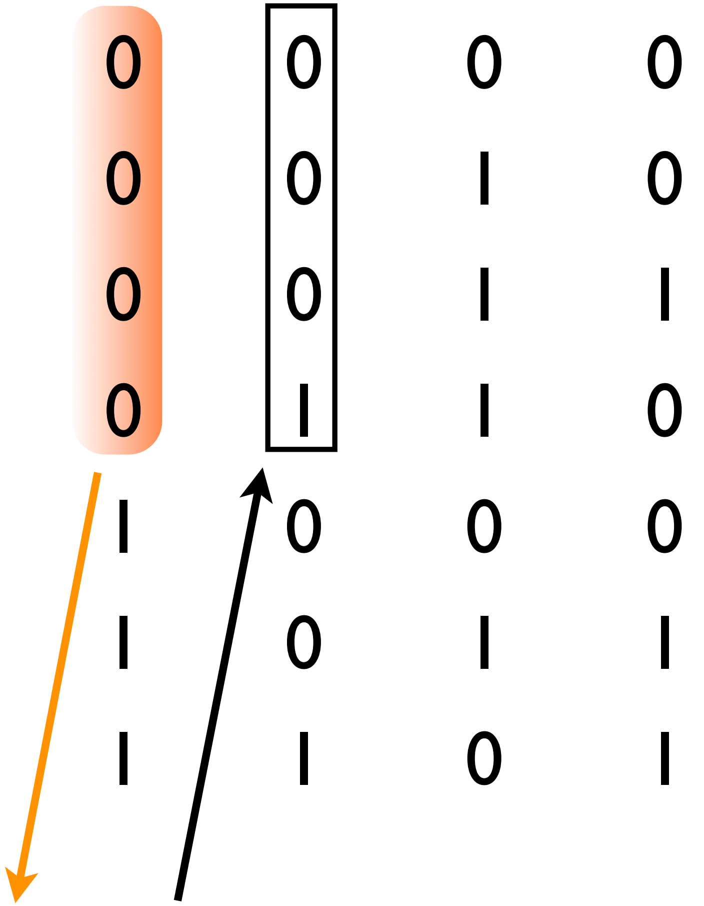
Context in bitwise recursive algorithm

- Represent elements as bitstrings (rows)

0	0	0	0
0	0	1	0
0	0	1	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1

Context in bitwise recursive algorithm

- Represent elements as bitstrings (rows)
- Encode from most to least significant bit
- Context obtained from more significant bits

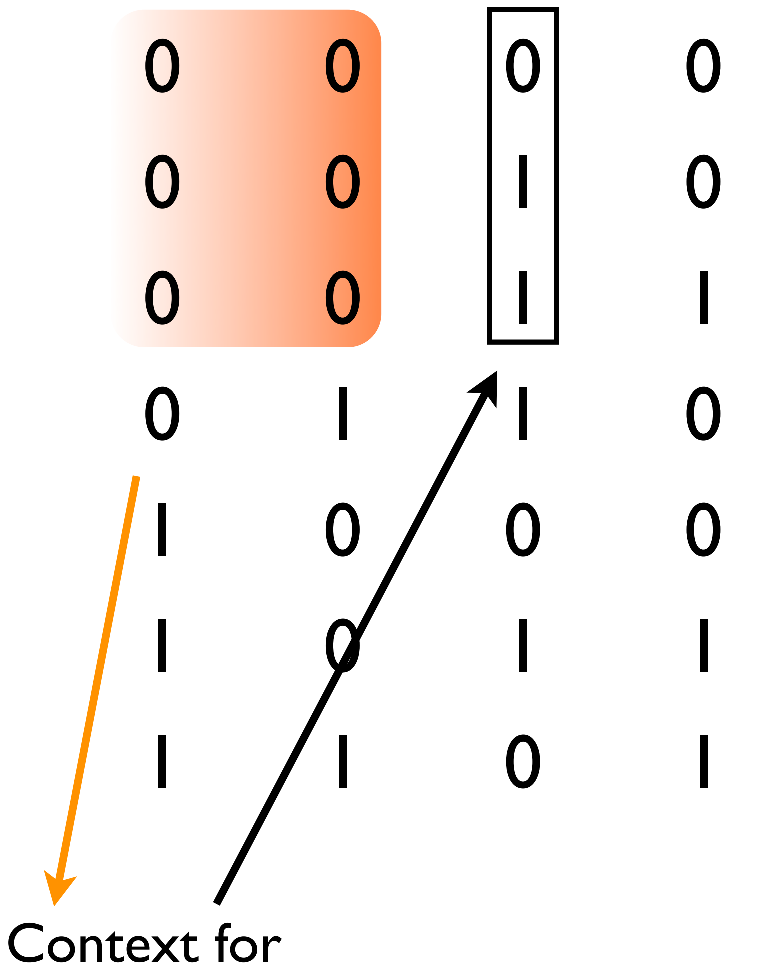


0 0 0 0
0 0 1 0
0 0 1 1
0 1 1 0
1 0 0 0
1 0 1 1
1 1 0 1

Context for

Context in bitwise recursive algorithm

- Represent elements as bitstrings (rows)
- Encode from most to least significant bit
- Context obtained from more significant bits



0	0	0	0
0	0	1	0
0	0	1	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1

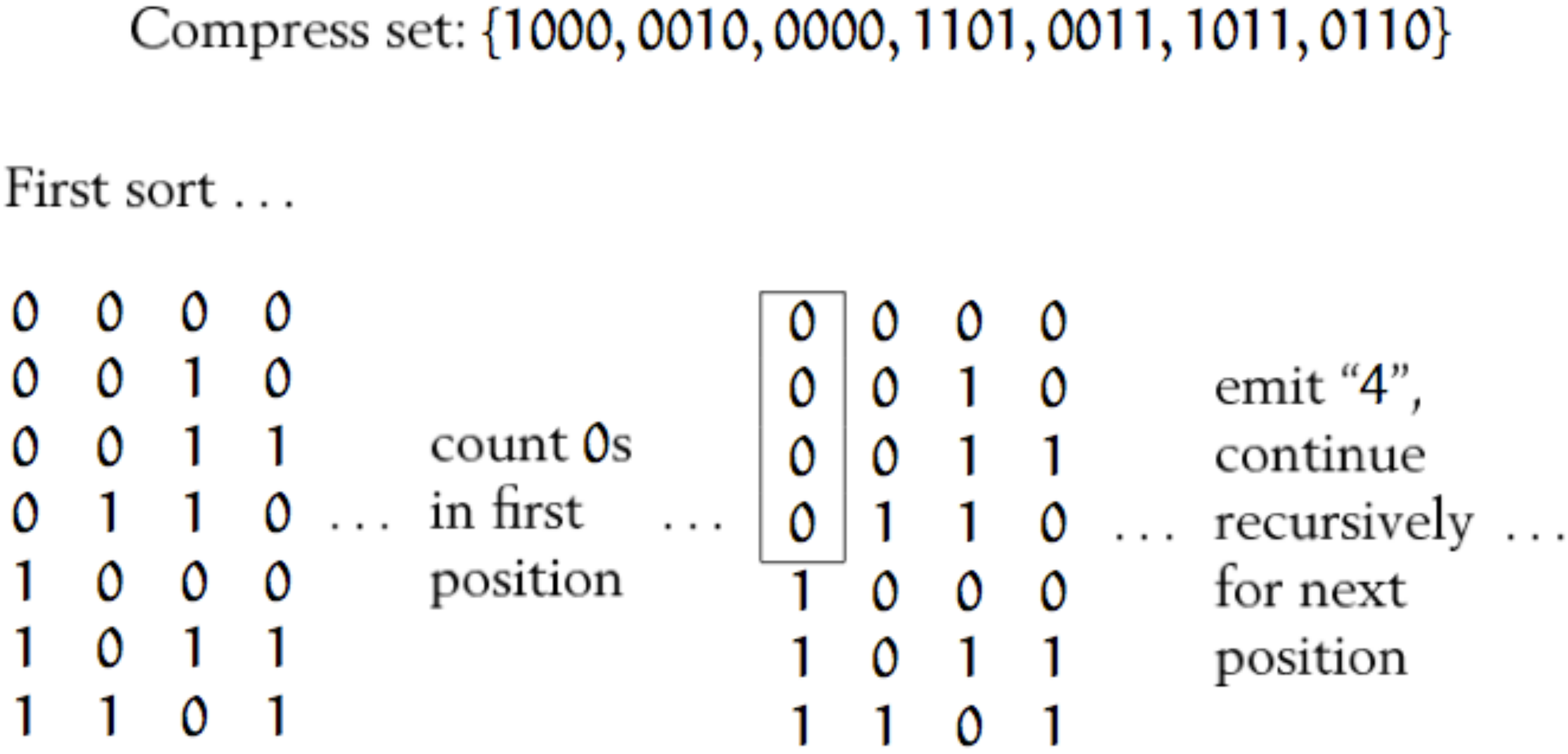
Known method 3: Interpolative coding

- Set: { 4, 9, 11, 14, 16, 17, 20, 21 }
- Encode 21 in range [0, |U|)
  14 in range [0, 21)
  9 in range [0, 14)
  4 in range [0, 9)
  11 in range (9, 14)
  17 in range (14, 21)
  16 in range (14, 17) *binary*
  20 in range (17, 21)

Moffat and Stuiver 2000

- (Simplified. Can also use known no of elements in range)

New method: recursive bitstring set encoding

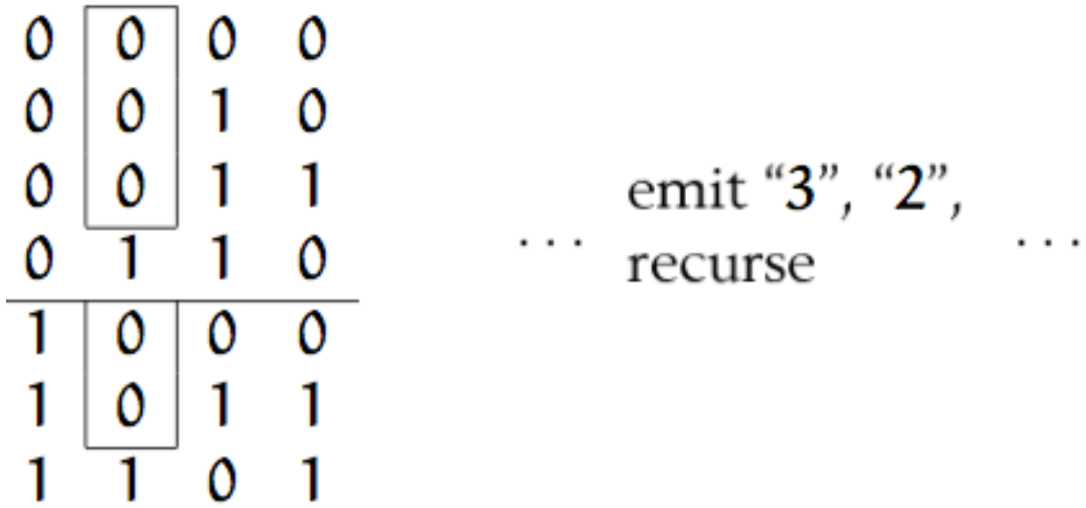


Compress set: $\{1000, 0010, 0000, 1101, 0011, 1011, 0110\}$

First sort …

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{array} \quad \ldots \quad \begin{array}{l} \text{count 0s} \\ \text{in first} \\ \text{position} \end{array} \quad \ldots \quad \begin{array}{|c|ccc} \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{array} \quad \ldots \quad \begin{array}{l} \text{emit “4”,} \\ \text{continue} \\ \text{recursively} \\ \text{for next} \\ \text{position} \end{array} \quad \ldots$$

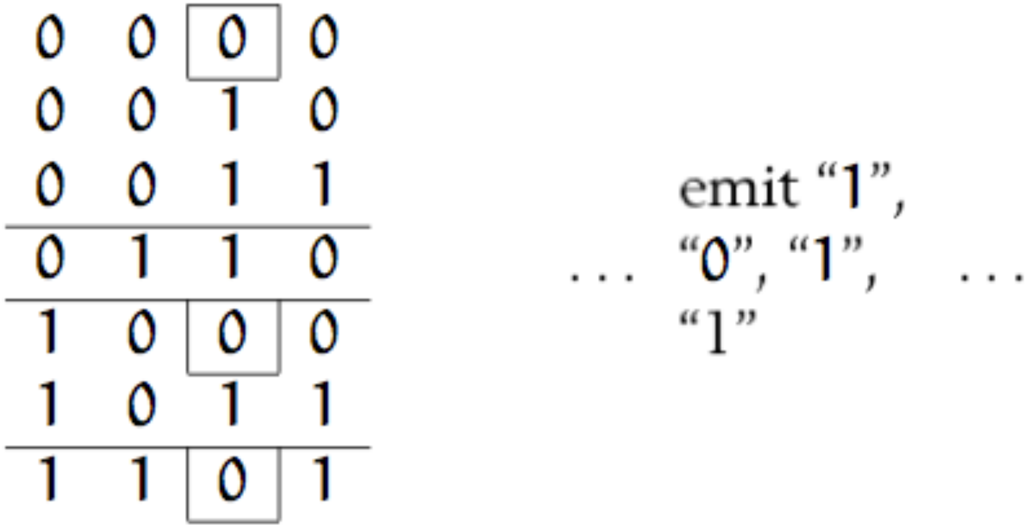
New method: recursive bitstring set encoding



0	0	0	0
0	0	1	0
0	0	1	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1

… emit “3”, “2”, recurse …

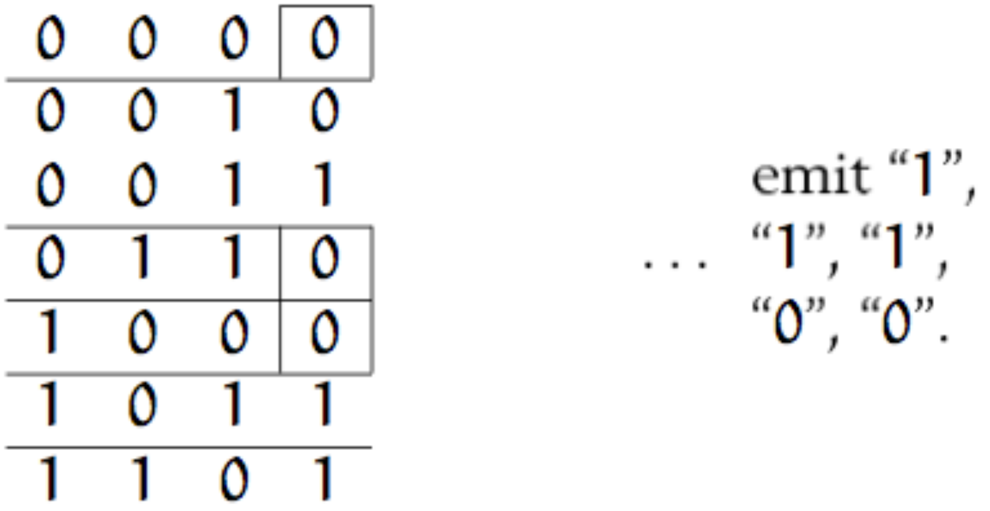
New method: recursive bitstring set encoding



0	0	$\boxed{0}$	0
0	0	1	0
0	0	1	1
0	1	1	0
1	0	$\boxed{0}$	0
1	0	1	1
1	1	$\boxed{0}$	1

emit “1”,
… “0”, “1”, …
“1”

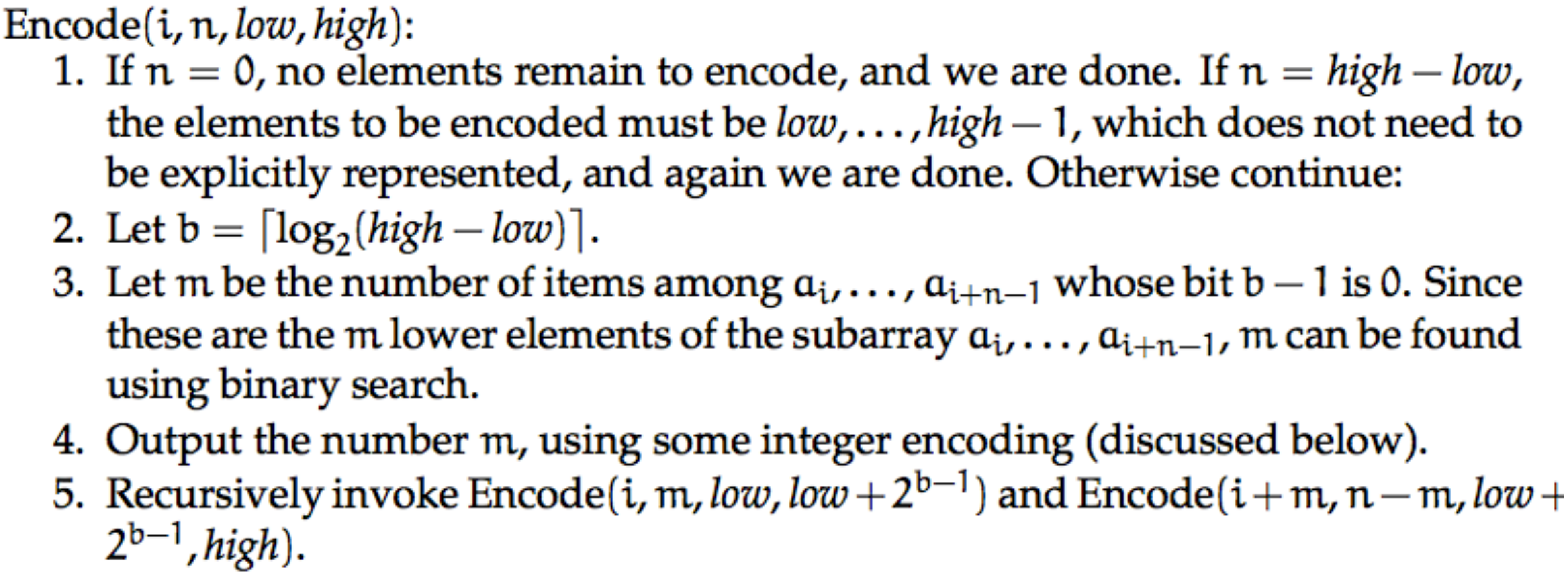
New method: recursive bitstring set encoding



0	0	0	0
0	0	1	0
0	0	1	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1

emit “1”,
. . . “1”, “1”,
“0”, “0”.

New method: recursive bitstring set encoding

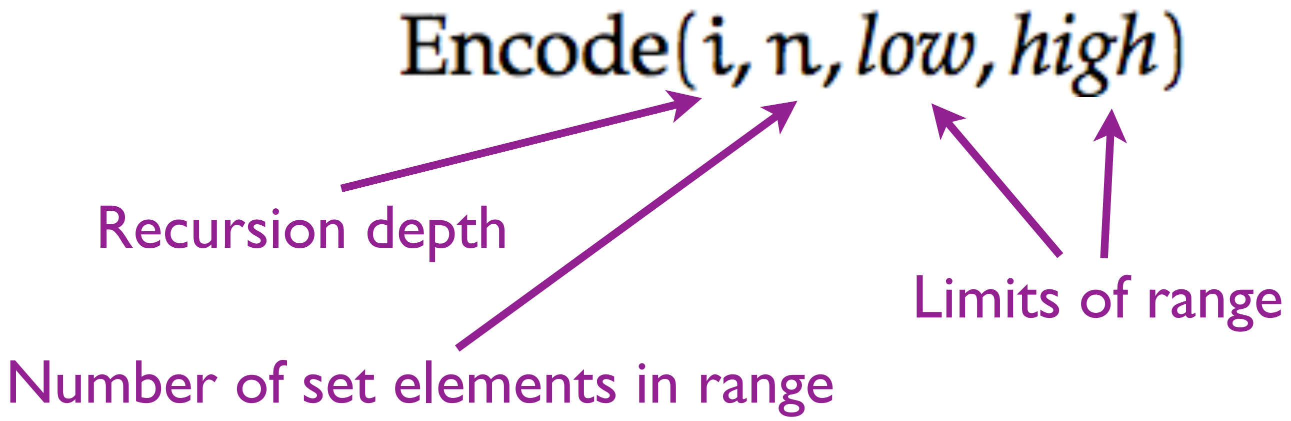


Encode($i, n, \mathit{low}, \mathit{high}$):

1. If $n = 0$, no elements remain to encode, and we are done. If $n = \mathit{high} - \mathit{low}$, the elements to be encoded must be $\mathit{low}, \ldots, \mathit{high} - 1$, which does not need to be explicitly represented, and again we are done. Otherwise continue:
2. Let $b = \lceil \log_2(\mathit{high} - \mathit{low}) \rceil$.
3. Let m be the number of items among $a_i, \ldots, a_{i+n-1}$ whose bit $b - 1$ is 0. Since these are the m lower elements of the subarray $a_i, \ldots, a_{i+n-1}$, m can be found using binary search.
4. Output the number m, using some integer encoding (discussed below).
5. Recursively invoke Encode($i, m, \mathit{low}, \mathit{low} + 2^{b-1}$) and Encode($i + m, n - m, \mathit{low} + 2^{b-1}, \mathit{high}$).

Encoding step

Emit number in smaller range as recursion deepens (~ interpolative)



$$\mathrm{Encode}(i, n, low, high)$$

Recursion depth

Number of set elements in range

Limits of range

Base: $\mathrm{Encode}(1, |S|, 0, |U|)$

Baseline: uniform element probabilities (no context)

$$s = 2^{\lceil \log_2(\mathit{high} - \mathit{low}) \rceil - 1}$$

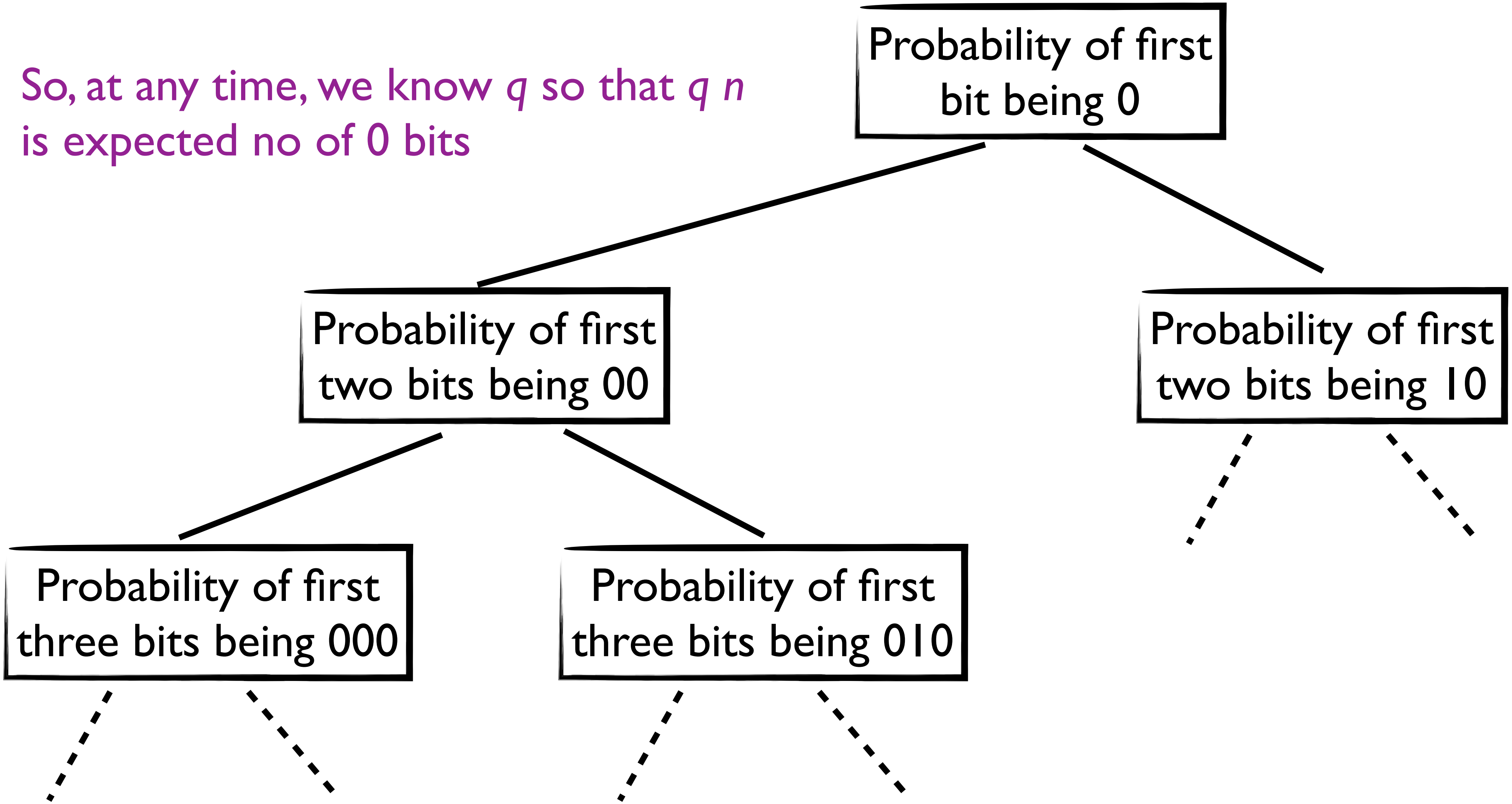
$$f = \mathit{high} - \mathit{low} - s$$

Hypergeometric distribution

$$\Pr(\text{elements starting with } 0 \text{ is } m) = \frac{\binom{s}{m}\binom{f}{n-m}}{\binom{s+f}{n}}$$

Context

So, at any time, we know *q* so that *q n* is expected no of 0 bits



Probability of first bit being 0

Probability of first two bits being 00

Probability of first two bits being 10

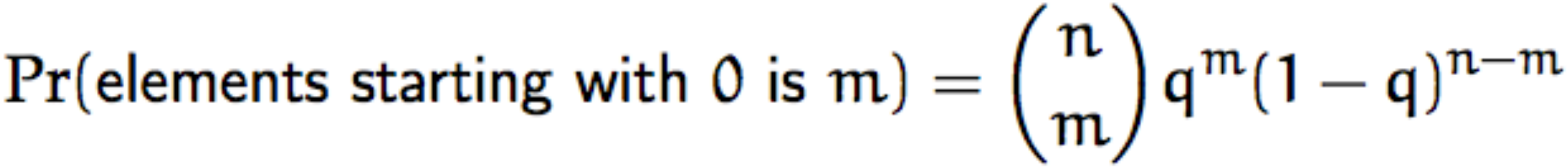
Probability of first three bits being 000

Probability of first three bits being 010

Binomial approximation

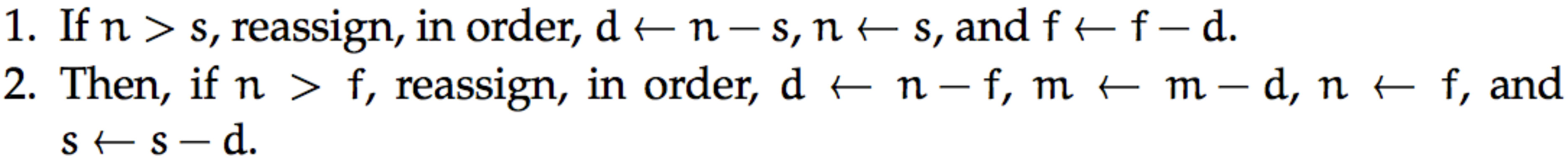
Estimate as if draw were *with replacement*, close if *s+f* is large in relation to *n*.

Binomial distribution



$$\Pr(\text{elements starting with 0 is } m) = \binom{n}{m} q^m (1-q)^{n-m}$$

Case exclusion: getting rid of nonzero probability for *m* > *s* and *m* < *n* – *f*



1. If $n > s$, reassign, in order, $d \leftarrow n - s$, $n \leftarrow s$, and $f \leftarrow f - d$.
2. Then, if $n > f$, reassign, in order, $d \leftarrow n - f$, $m \leftarrow m - d$, $n \leftarrow f$, and $s \leftarrow s - d$.

Hypergeometric rescaled

$$\frac{s}{s+f} = q$$

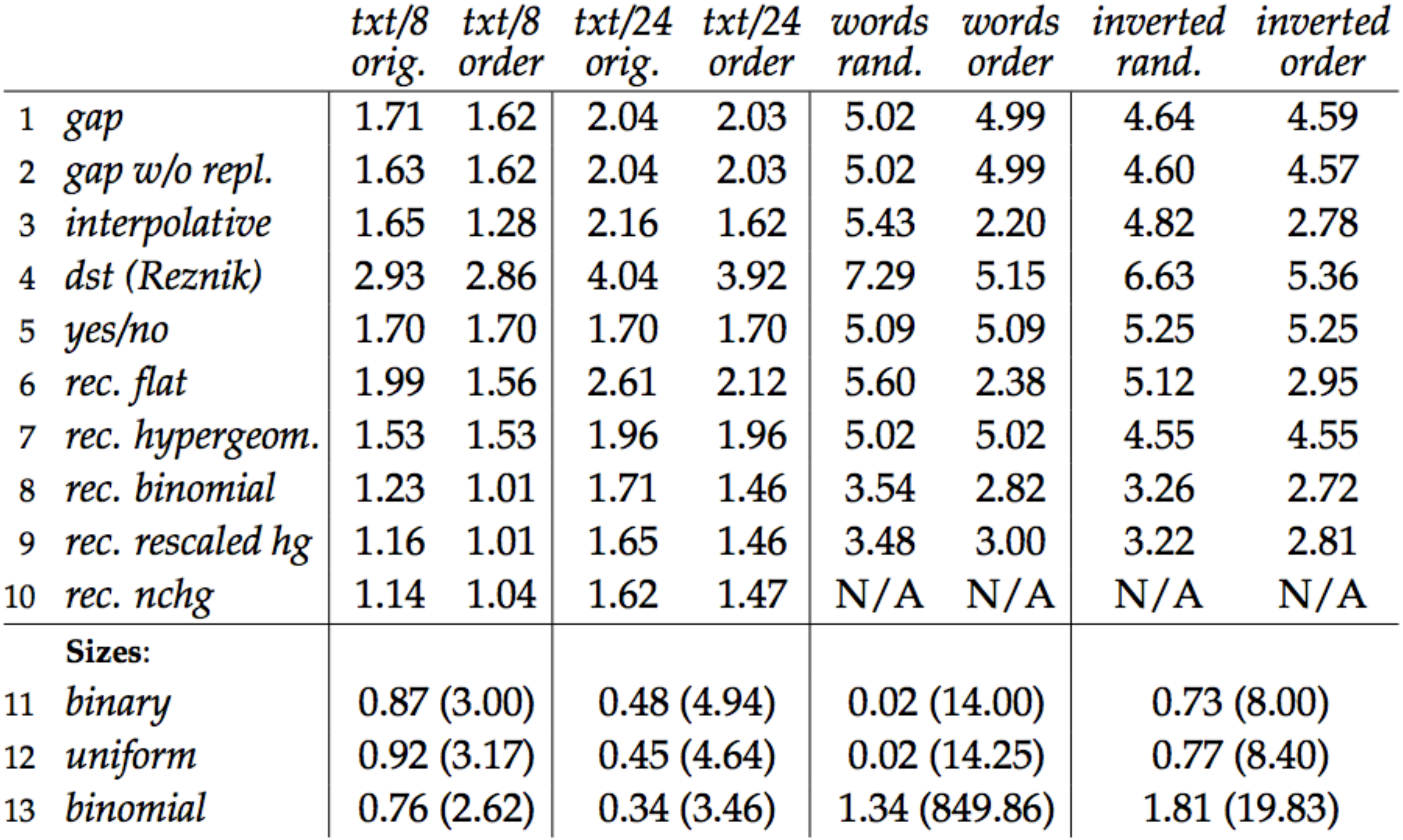
If $s/f \geq q/(1-q)$, reassign $f \leftarrow [s(1-q)/q]$

If $s/f < q/(1-q)$, reassign $s \leftarrow [fq/(1-q)]$

Non-central hypergeometric

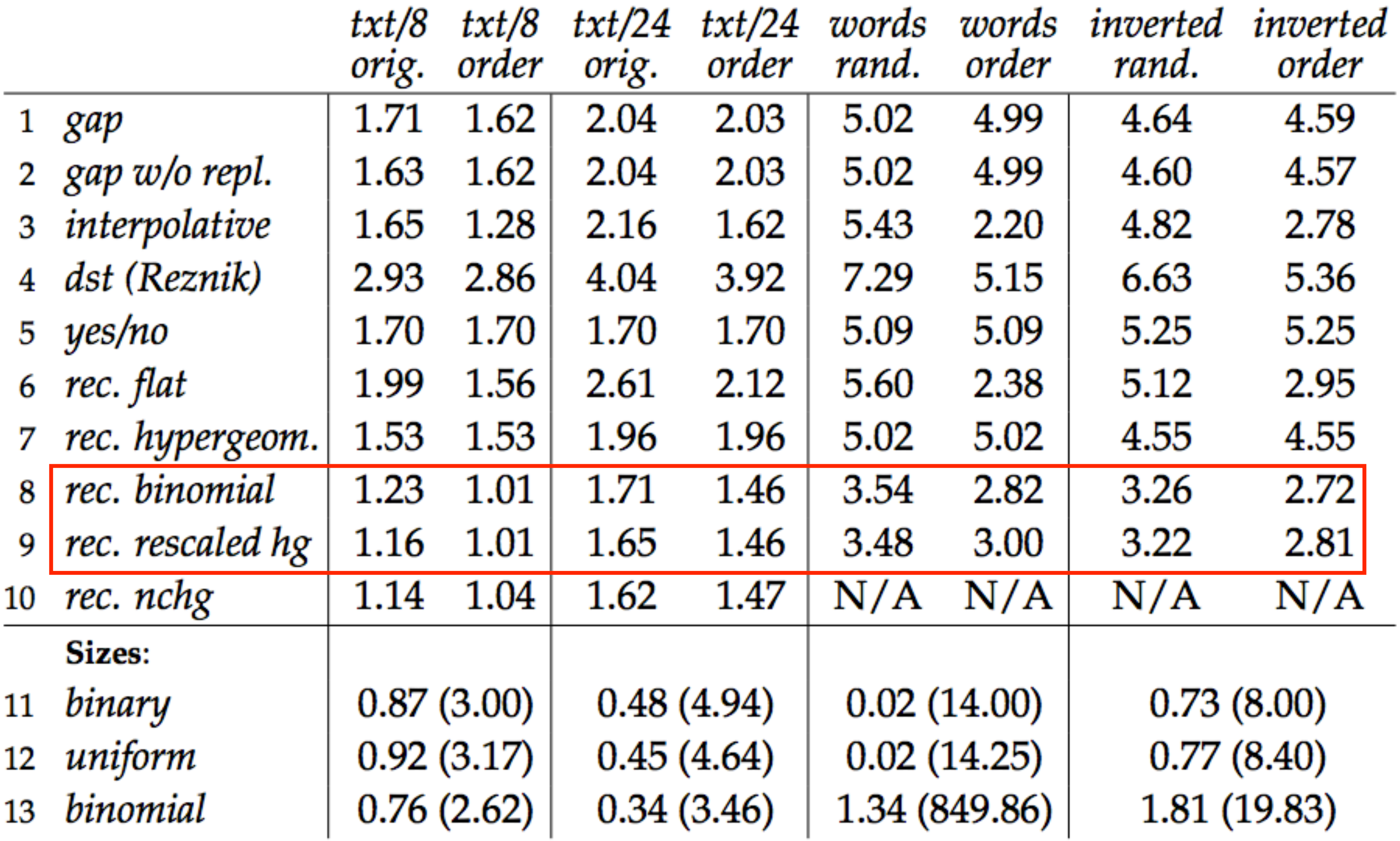
- Introduce a weight $w = \frac{f}{s} \cdot \frac{q}{1-q}$
- Wallenius' non-central hypergeometric distribution

Results



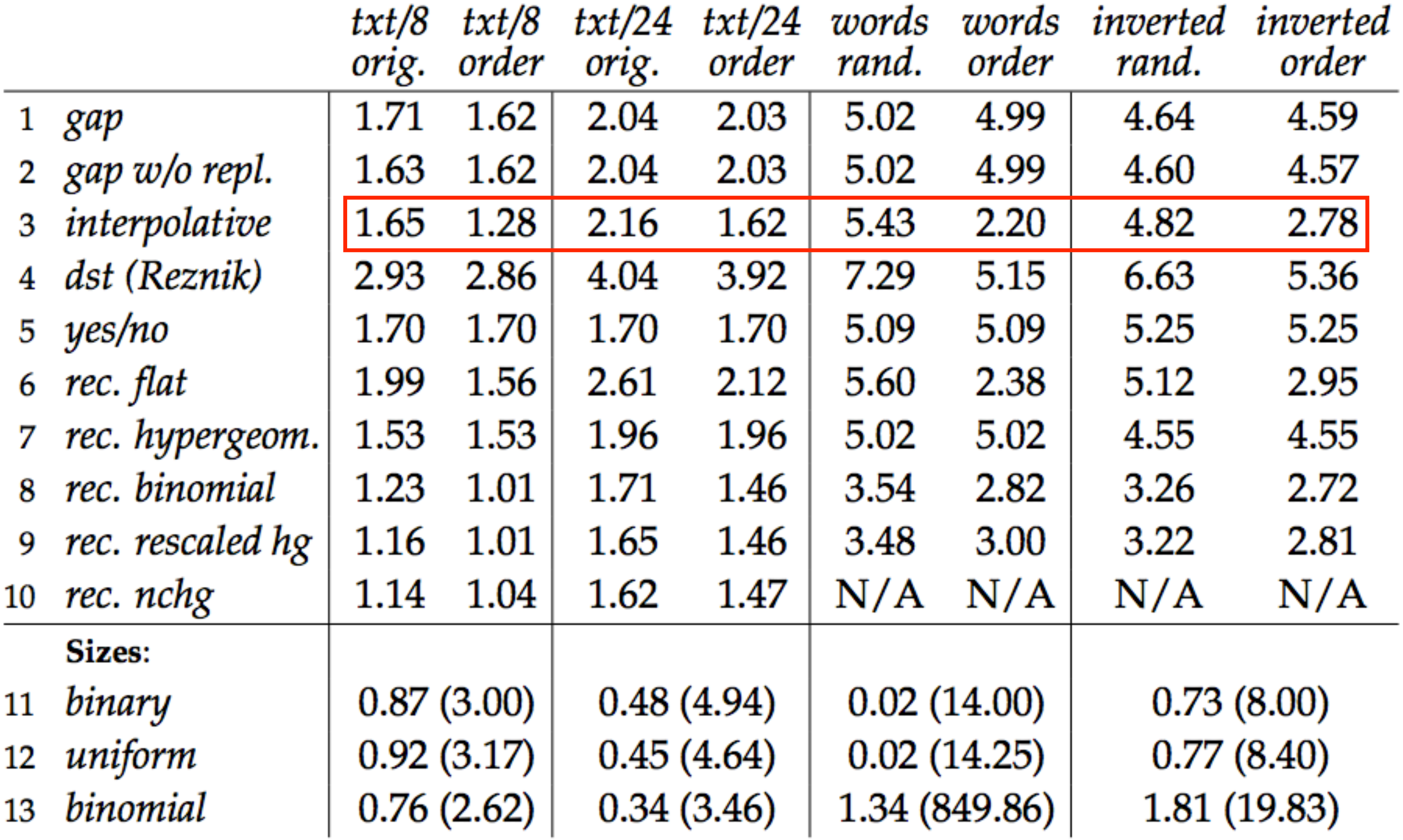
		txt/8 orig.	txt/8 order	txt/24 orig.	txt/24 order	words rand.	words order	inverted rand.	inverted order
1	*gap*	1.71	1.62	2.04	2.03	5.02	4.99	4.64	4.59
2	*gap w/o repl.*	1.63	1.62	2.04	2.03	5.02	4.99	4.60	4.57
3	*interpolative*	1.65	1.28	2.16	1.62	5.43	2.20	4.82	2.78
4	*dst (Reznik)*	2.93	2.86	4.04	3.92	7.29	5.15	6.63	5.36
5	*yes/no*	1.70	1.70	1.70	1.70	5.09	5.09	5.25	5.25
6	*rec. flat*	1.99	1.56	2.61	2.12	5.60	2.38	5.12	2.95
7	*rec. hypergeom.*	1.53	1.53	1.96	1.96	5.02	5.02	4.55	4.55
8	*rec. binomial*	1.23	1.01	1.71	1.46	3.54	2.82	3.26	2.72
9	*rec. rescaled hg*	1.16	1.01	1.65	1.46	3.48	3.00	3.22	2.81
10	*rec. nchg*	1.14	1.04	1.62	1.47	N/A	N/A	N/A	N/A
	Sizes:								
11	*binary*	0.87 (3.00)		0.48 (4.94)		0.02 (14.00)		0.73 (8.00)	
12	*uniform*	0.92 (3.17)		0.45 (4.64)		0.02 (14.25)		0.77 (8.40)	
13	*binomial*	0.76 (2.62)		0.34 (3.46)		1.34 (849.86)		1.81 (19.83)	

Results



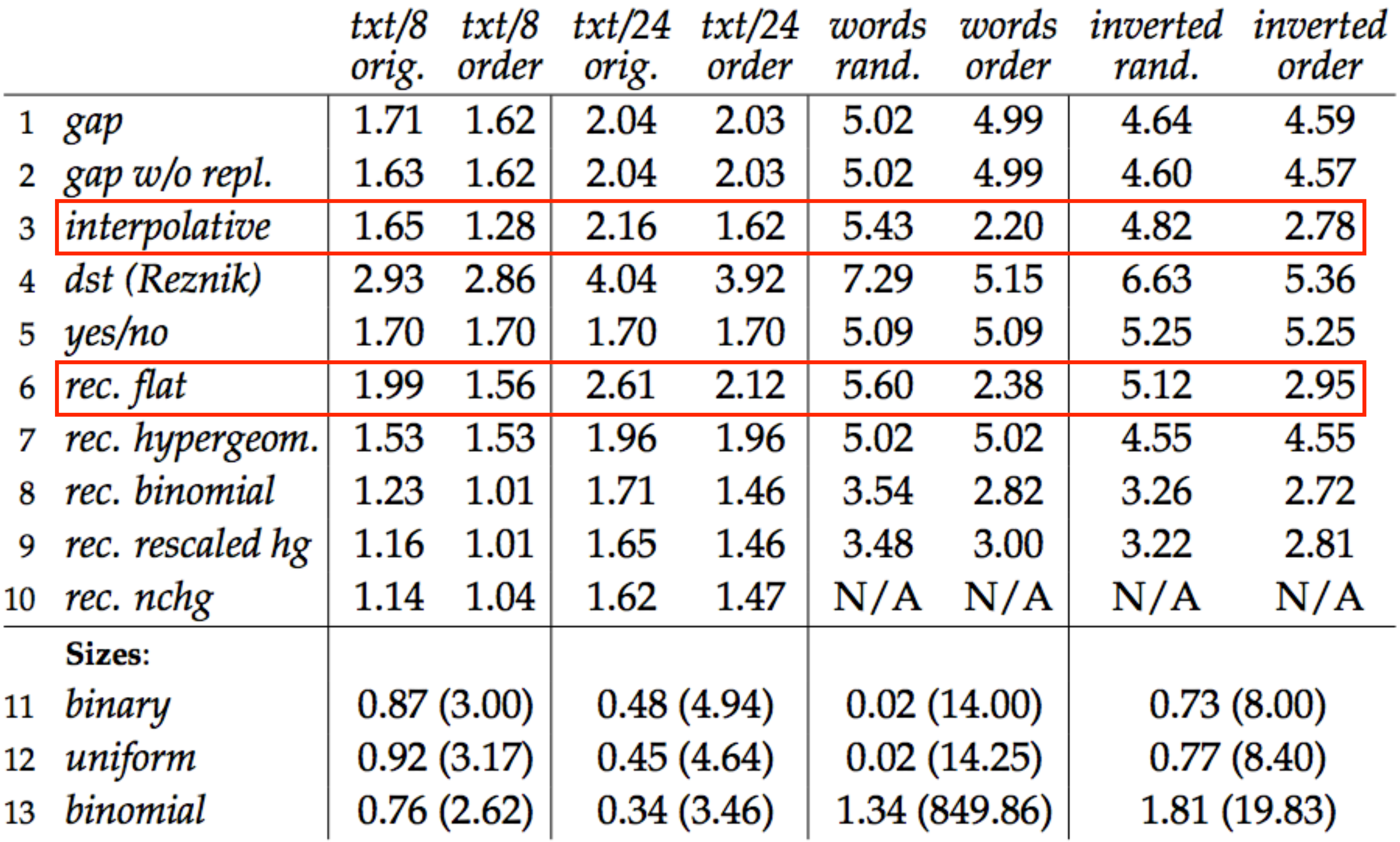
		txt/8 orig.	txt/8 order	txt/24 orig.	txt/24 order	words rand.	words order	inverted rand.	inverted order
1	*gap*	1.71	1.62	2.04	2.03	5.02	4.99	4.64	4.59
2	*gap w/o repl.*	1.63	1.62	2.04	2.03	5.02	4.99	4.60	4.57
3	*interpolative*	1.65	1.28	2.16	1.62	5.43	2.20	4.82	2.78
4	*dst (Reznik)*	2.93	2.86	4.04	3.92	7.29	5.15	6.63	5.36
5	*yes/no*	1.70	1.70	1.70	1.70	5.09	5.09	5.25	5.25
6	*rec. flat*	1.99	1.56	2.61	2.12	5.60	2.38	5.12	2.95
7	*rec. hypergeom.*	1.53	1.53	1.96	1.96	5.02	5.02	4.55	4.55
8	*rec. binomial*	1.23	1.01	1.71	1.46	3.54	2.82	3.26	2.72
9	*rec. rescaled hg*	1.16	1.01	1.65	1.46	3.48	3.00	3.22	2.81
10	*rec. nchg*	1.14	1.04	1.62	1.47	N/A	N/A	N/A	N/A
	Sizes:								
11	*binary*	0.87 (3.00)		0.48 (4.94)		0.02 (14.00)		0.73 (8.00)	
12	*uniform*	0.92 (3.17)		0.45 (4.64)		0.02 (14.25)		0.77 (8.40)	
13	*binomial*	0.76 (2.62)		0.34 (3.46)		1.34 (849.86)		1.81 (19.83)	

Results



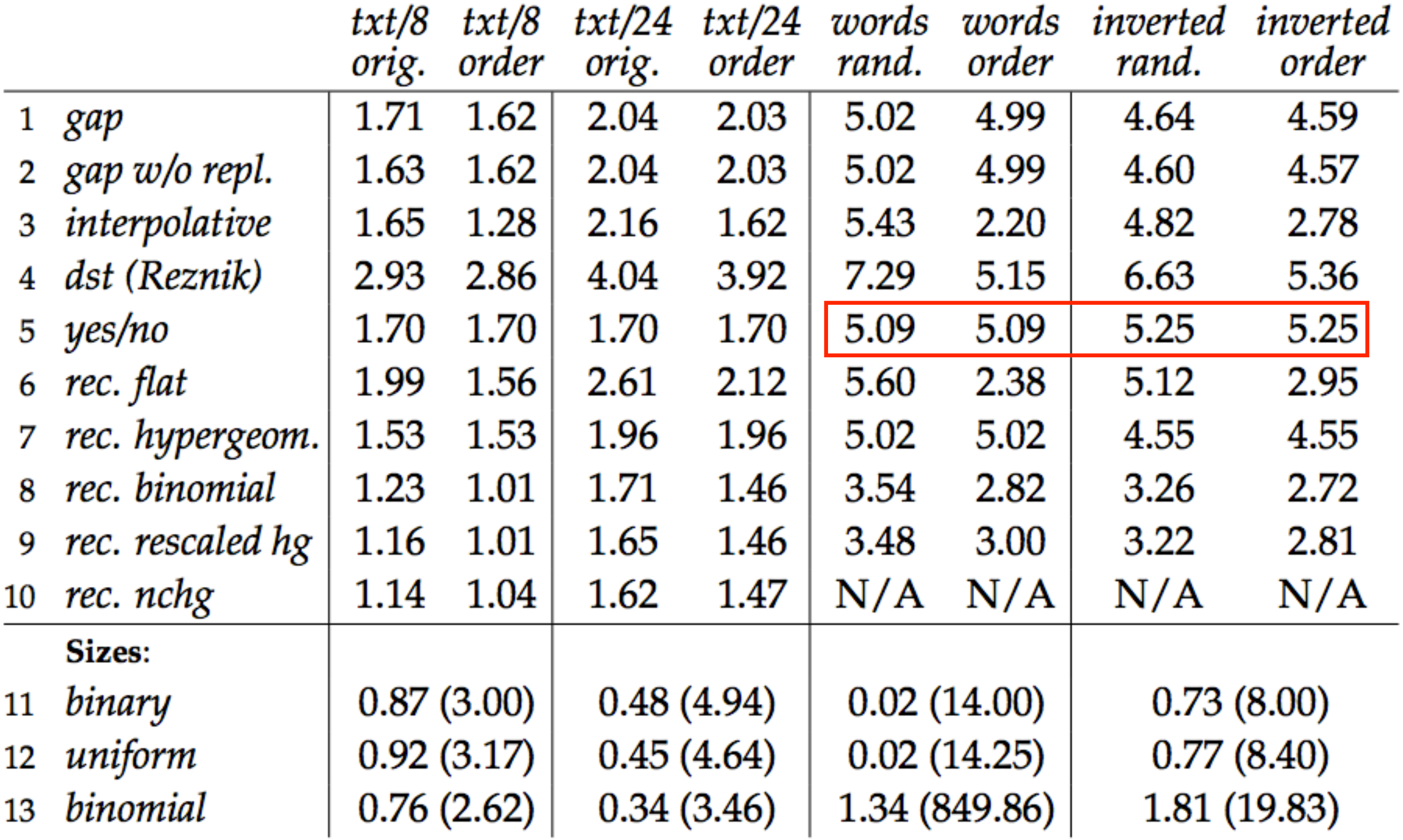
		txt/8 orig.	txt/8 order	txt/24 orig.	txt/24 order	words rand.	words order	inverted rand.	inverted order
1	*gap*	1.71	1.62	2.04	2.03	5.02	4.99	4.64	4.59
2	*gap w/o repl.*	1.63	1.62	2.04	2.03	5.02	4.99	4.60	4.57
3	*interpolative*	1.65	1.28	2.16	1.62	5.43	2.20	4.82	2.78
4	*dst (Reznik)*	2.93	2.86	4.04	3.92	7.29	5.15	6.63	5.36
5	*yes/no*	1.70	1.70	1.70	1.70	5.09	5.09	5.25	5.25
6	*rec. flat*	1.99	1.56	2.61	2.12	5.60	2.38	5.12	2.95
7	*rec. hypergeom.*	1.53	1.53	1.96	1.96	5.02	5.02	4.55	4.55
8	*rec. binomial*	1.23	1.01	1.71	1.46	3.54	2.82	3.26	2.72
9	*rec. rescaled hg*	1.16	1.01	1.65	1.46	3.48	3.00	3.22	2.81
10	*rec. nchg*	1.14	1.04	1.62	1.47	N/A	N/A	N/A	N/A
	Sizes:								
11	*binary*	0.87 (3.00)		0.48 (4.94)		0.02 (14.00)		0.73 (8.00)	
12	*uniform*	0.92 (3.17)		0.45 (4.64)		0.02 (14.25)		0.77 (8.40)	
13	*binomial*	0.76 (2.62)		0.34 (3.46)		1.34 (849.86)		1.81 (19.83)	

Results



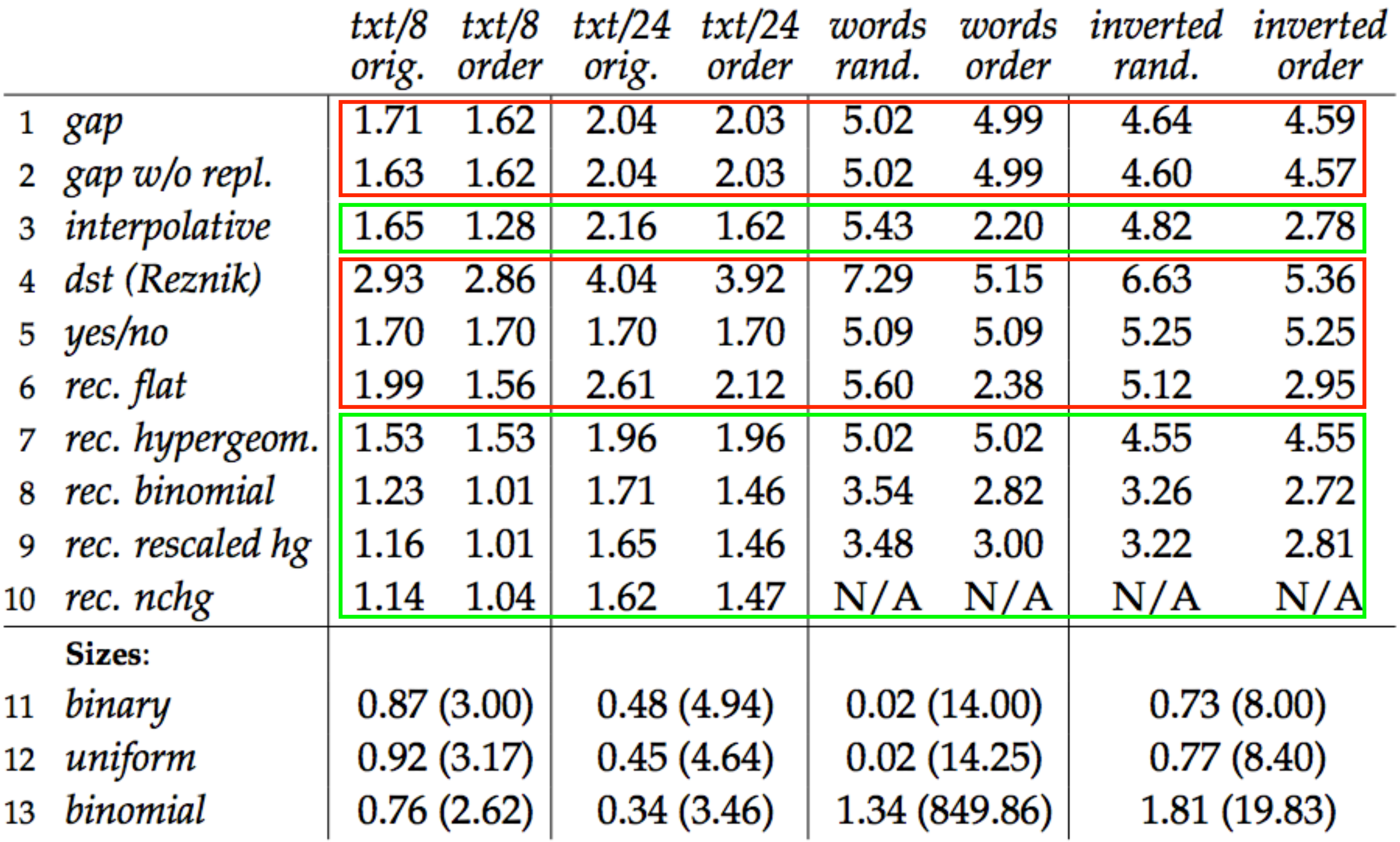
		txt/8 orig.	txt/8 order	txt/24 orig.	txt/24 order	words rand.	words order	inverted rand.	inverted order
1	*gap*	1.71	1.62	2.04	2.03	5.02	4.99	4.64	4.59
2	*gap w/o repl.*	1.63	1.62	2.04	2.03	5.02	4.99	4.60	4.57
3	*interpolative*	1.65	1.28	2.16	1.62	5.43	2.20	4.82	2.78
4	*dst (Reznik)*	2.93	2.86	4.04	3.92	7.29	5.15	6.63	5.36
5	*yes/no*	1.70	1.70	1.70	1.70	5.09	5.09	5.25	5.25
6	*rec. flat*	1.99	1.56	2.61	2.12	5.60	2.38	5.12	2.95
7	*rec. hypergeom.*	1.53	1.53	1.96	1.96	5.02	5.02	4.55	4.55
8	*rec. binomial*	1.23	1.01	1.71	1.46	3.54	2.82	3.26	2.72
9	*rec. rescaled hg*	1.16	1.01	1.65	1.46	3.48	3.00	3.22	2.81
10	*rec. nchg*	1.14	1.04	1.62	1.47	N/A	N/A	N/A	N/A
	Sizes:								
11	*binary*	0.87 (3.00)		0.48 (4.94)		0.02 (14.00)		0.73 (8.00)	
12	*uniform*	0.92 (3.17)		0.45 (4.64)		0.02 (14.25)		0.77 (8.40)	
13	*binomial*	0.76 (2.62)		0.34 (3.46)		1.34 (849.86)		1.81 (19.83)	

Results



		txt/8 *orig.*	*txt/8* *order*	*txt/24* *orig.*	*txt/24* *order*	*words* *rand.*	*words* *order*	*inverted* *rand.*	*inverted* *order*
1	*gap*	1.71	1.62	2.04	2.03	5.02	4.99	4.64	4.59
2	*gap w/o repl.*	1.63	1.62	2.04	2.03	5.02	4.99	4.60	4.57
3	*interpolative*	1.65	1.28	2.16	1.62	5.43	2.20	4.82	2.78
4	*dst (Reznik)*	2.93	2.86	4.04	3.92	7.29	5.15	6.63	5.36
5	*yes/no*	1.70	1.70	1.70	1.70	5.09	5.09	5.25	5.25
6	*rec. flat*	1.99	1.56	2.61	2.12	5.60	2.38	5.12	2.95
7	*rec. hypergeom.*	1.53	1.53	1.96	1.96	5.02	5.02	4.55	4.55
8	*rec. binomial*	1.23	1.01	1.71	1.46	3.54	2.82	3.26	2.72
9	*rec. rescaled hg*	1.16	1.01	1.65	1.46	3.48	3.00	3.22	2.81
10	*rec. nchg*	1.14	1.04	1.62	1.47	N/A	N/A	N/A	N/A
	Sizes:								
11	*binary*	0.87 (3.00)		0.48 (4.94)		0.02 (14.00)		0.73 (8.00)	
12	*uniform*	0.92 (3.17)		0.45 (4.64)		0.02 (14.25)		0.77 (8.40)	
13	*binomial*	0.76 (2.62)		0.34 (3.46)		1.34 (849.86)		1.81 (19.83)	

Results



		txt/8 *orig.*	*txt/8* *order*	*txt/24* *orig.*	*txt/24* *order*	*words* *rand.*	*words* *order*	*inverted* *rand.*	*inverted* *order*
1	*gap*	1.71	1.62	2.04	2.03	5.02	4.99	4.64	4.59
2	*gap w/o repl.*	1.63	1.62	2.04	2.03	5.02	4.99	4.60	4.57
3	*interpolative*	1.65	1.28	2.16	1.62	5.43	2.20	4.82	2.78
4	*dst (Reznik)*	2.93	2.86	4.04	3.92	7.29	5.15	6.63	5.36
5	*yes/no*	1.70	1.70	1.70	1.70	5.09	5.09	5.25	5.25
6	*rec. flat*	1.99	1.56	2.61	2.12	5.60	2.38	5.12	2.95
7	*rec. hypergeom.*	1.53	1.53	1.96	1.96	5.02	5.02	4.55	4.55
8	*rec. binomial*	1.23	1.01	1.71	1.46	3.54	2.82	3.26	2.72
9	*rec. rescaled hg*	1.16	1.01	1.65	1.46	3.48	3.00	3.22	2.81
10	*rec. nchg*	1.14	1.04	1.62	1.47	N/A	N/A	N/A	N/A
	Sizes:								
11	*binary*	0.87 (3.00)		0.48 (4.94)		0.02 (14.00)		0.73 (8.00)	
12	*uniform*	0.92 (3.17)		0.45 (4.64)		0.02 (14.25)		0.77 (8.40)	
13	*binomial*	0.76 (2.62)		0.34 (3.46)		1.34 (849.86)		1.81 (19.83)	

Thank you.