Analysing New Entropy Measures for Tries

32nd International Symposium on String Processing and Information Retrieval

Lorenzo Carfagna[1] and **Carlo Tosoni**[2]

1. University of Pisa, Italy
2. Ca' Foscari University of Venice, Italy

City St George's, University of London

10/09/2025

The Worst-Case Entropy

Definition: Worst-Case Entropy

Let $\mathcal{U}$ be a set, the **worst-case entropy** $\mathcal{H}^{wc}(\mathcal{U})$ of $\mathcal{U}$ is defined as

$$\mathcal{H}^{wc}(\mathcal{U}) = \log_2 |\mathcal{U}|$$

Example, if $\mathcal{U} = \{\text{dog, cat, bird, mouse}\}$, then $\mathcal{H}^{wc}(\mathcal{U}) = \log_2 |\mathcal{U}| = \log_2 4 = \mathbf{2}$

The Worst-Case Entropy of a String

Consider the string **S** = **aaaaaabaaaaaaaaaabaaa**.

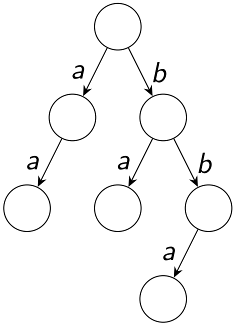
- If we consider $\mathcal{U}$ as **the set of strings** of **length** $n = \mathbf{20}$ over an **alphabet of size** $\sigma = \mathbf{2}$, then:

$$\mathcal{H}^{wc}(\mathcal{U}) = n \log \sigma = 20 \text{ bits}$$

- If $\mathcal{U}$ is the set of strings where **a** and **b appear 18** and **2 times**:

$$\mathcal{H}^{wc}(\mathcal{U}) = \log \binom{20}{2} \approx 7.57 \text{ bits}$$

The Worst-Case Entropy of a Trie



There exists a **famous worst-case formula** for the **set of tries** having **n** nodes over an **alphabet of size** σ.

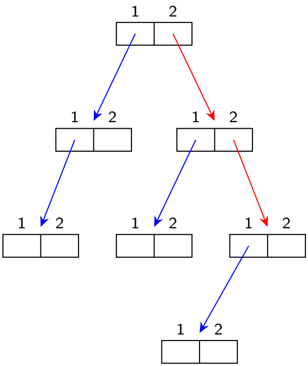
$$\mathcal{H}^{wc}(\mathcal{U}) = \log \frac{1}{n}\binom{n\sigma}{n-1} \quad [1]$$

Ex. if $n = 7$ and $\sigma = 2$, then $\log \frac{1}{7}\binom{14}{6} \approx 8.7$ bits

What if we consider tries with a **given symbol distribution**?

1. *R. Graham, D. Knuth, and O. Patashnik: Concrete Mathematics. Addison-Wesley. (1994)*

The Worst-Case Entropy of a Trie



The number of t-ary trees with a **fixed number of first, second, . . ., t-th children** was computed using generating functions [2].

Ex. the 2-ary on the left has **4 first children** and **2 second children**.

- **In bijection** with our class of tries.
- $|\mathcal{U}| = \frac{1}{n} \prod_{c \in \Sigma} \binom{n}{n_c}$,

$n_c = \#$ edges labeled by the character c.

2. H. Prodinger. Counting edges according to edge-type in t-ary trees. arXiv. (2022)

Our contributions

1. Provide an **alternative proof** for the formula $|\mathcal{U}| = \frac{1}{n} \prod_{c \in \Sigma} \binom{n}{n_c}$

   By using a simple bijection!

2. Introducing corresponding **worst-case entropy** $\mathcal{H}^{wc}(\mathcal{U})$

3. Introducing an **empirical entropy for tries** $\mathcal{H}_k(\mathcal{T})$

4. **Compress** and **index** a trie in $n\mathcal{H}_k(\mathcal{T}) + o(n)$ bits using the **XBWT**

The Function $f : \mathcal{U} \to \mathcal{M}$

Domain: $\mathcal{U} \leftarrow$ set of tries having $\mathbf{n_c}$ **edges** labeled by $c \in \Sigma$

Codomain: $\mathcal{M} \leftarrow$ set of $\sigma \times n$ **binary matrices** having $\mathbf{n_c}$ **ones** at row c.

$\xrightarrow{\mathbf{f}}$

	u_1	u_2	u_3	u_4	u_5
a	1	0	1	0	0
b	1	0	0	0	0
c	0	0	1	0	0

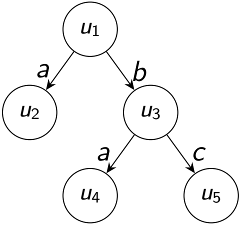
To compute the matrix $M = f(\mathcal{T})$:

1. **Sort the nodes** of $\mathcal{T}$ based on a **pre-order visit**. $(u_1, u_2, u_3, u_4, u_5$ in fig.)
2. Set $\mathbf{M[i][c] = 1}$ **iff** there exists the edge $\mathbf{u_i} \xrightarrow{\mathbf{c}} \mathbf{v}$.

The Function $f : \mathcal{U} \to \mathcal{M}$

Domain: $\mathcal{U} \leftarrow$ set of tries having $\mathbf{n_c}$ **edges** labeled by $c \in \Sigma$

Codomain: $\mathcal{M} \leftarrow$ set of $\sigma \times n$ **binary matrices** having $\mathbf{n_c}$ **ones** at row c.



$\xrightarrow{\mathbf{f}}$

	u_1	u_2	u_3	u_4	u_5
a	1	0	1	0	0
b	1	0	0	0	0
c	0	0	1	0	0

To compute the matrix $M = f(\mathcal{T})$:

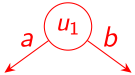
1. **Sort the nodes** of $\mathcal{T}$ based on a **pre-order visit**. (u_1, u_2, u_3, u_4, u_5 in fig.)
2. Set $\mathbf{M[i][c] = 1}$ **iff** there exists the edge $\mathbf{u_i} \xrightarrow{\mathbf{c}} \mathbf{v}$.

Inverting Function f

The function f is **injective**, but **not surijective**: some matrices in $\mathcal{M}$ do not correspond to any trie.

	u_1	u_2	u_3	u_4	u_5
a	1	0	0	1	0
b	1	0	0	0	0
c	0	0	0	0	1

$\xrightarrow{\mathbf{f}^{-1}}$



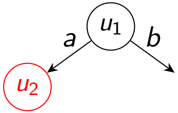
Connectivity constraints could be violated during the inversion process.

Inverting Function f

The function f is **injective**, but **not surijective**: some matrices in $\mathcal{M}$ do not correspond to any trie.

	u_1	u_2	u_3	u_4	u_5
a	1	0	0	1	0
b	1	0	0	0	0
c	0	0	0	0	1

$$\xrightarrow{\mathbf{f}^{-1}}$$



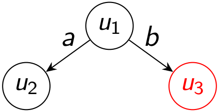
Connectivity constraints could be violated during the inversion process.

Inverting Function f

The function f is **injective**, but **not surijective**: some matrices in $\mathcal{M}$ do not correspond to any trie.

	u_1	u_2	u_3	u_4	u_5
a	1	0	0	1	0
b	1	0	0	0	0
c	0	0	0	0	1

$$\xrightarrow{\mathbf{f}^{-1}}$$



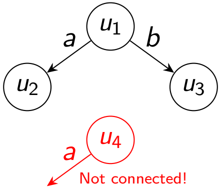
Connectivity constraints could be violated during the inversion process.

Inverting Function f

The function f is **injective**, but **not surijective**: some matrices in $\mathcal{M}$ do not correspond to any trie.

	u_1	u_2	u_3	u_4	u_5
a	1	0	0	1	0
b	1	0	0	0	0
c	0	0	0	0	1

$\xrightarrow{\mathbf{f}^{-1}}$



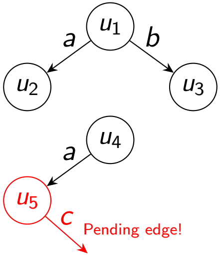
Connectivity constraints could be violated during the inversion process.

Inverting Function f

The function f is **injective**, but **not surijective**: some matrices in $\mathcal{M}$ do not correspond to any trie.

	u_1	u_2	u_3	u_4	u_5
a	1	0	0	1	0
b	1	0	0	0	0
c	0	0	0	0	1

$\xrightarrow{\mathbf{f}^{-1}}$



Connectivity constraints could be violated during the inversion process.

Rotating the Matrix

What happens if we **rotate the matrix**?

	u_1	u_2	u_3	u_4	u_5
a	1	0	0	1	0
b	1	0	0	0	0
c	0	0	0	0	1

Rotating two columns!

	u_1	u_2	u_3	u_4	u_5
a	1	0	1	0	0
b	0	0	1	0	0
c	0	1	0	0	0

Now the **matrix is invertible**!

Rotating the Matrix

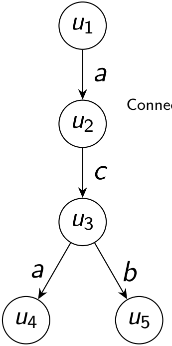
What happens if we **rotate the matrix**?

	u_1	u_2	u_3	u_4	u_5
a	1	0	0	1	0
b	1	0	0	0	0
c	0	0	0	0	1

Rotating

two columns!

	u_1	u_2	u_3	u_4	u_5
a	1	0	1	0	0
b	0	0	1	0	0
c	0	1	0	0	0



Connected ☺

Now the **matrix is invertible**!

New Worst-Case Formula Measure

That's not by chance! Using a result about integer sequences [3] we deduced:

1. Every matrix M in $\mathcal{M}$ has exactly **n distinct rotations**. $n = \#$ of columns
2. The **rotation of M that is invertible** exists and is unique.

We observe $|\mathcal{M}| = \prod_{c \in \Sigma} \binom{n}{n_c}$ $n_c =$ number of ones at the c-th row

Consequently, $|\mathcal{U}| = \frac{1}{n} \prod_{c \in \Sigma} \binom{n}{n_c}$ and $\mathcal{H}^{wc}(\mathcal{U}) = \sum_{c \in \Sigma} \log \binom{n}{n_c} - \log n$.

3. G. Rote. Binary trees with nodes having 0, 1, and 2 children. Séminaire Lotharingien de Combinatoire. (1997)

New Worst-Case Formula Measure

That's not by chance! Using a result about integer sequences [3] we deduced:

1. Every matrix M in $\mathcal{M}$ has exactly **n distinct rotations**. $n = \#$ of columns
2. The **rotation of M that is invertible** exists and is unique.

We observe $|\mathcal{M}| = \prod_{c \in \Sigma} \binom{n}{n_c}$ $n_c =$ number of ones at the c-th row

Consequently, $|\mathcal{U}| = \frac{1}{n} \prod_{c \in \Sigma} \binom{n}{n_c}$ and $\mathcal{H}^{wc}(\mathcal{U}) = \sum_{c \in \Sigma} \log \binom{n}{n_c} - \log n$.

3. G. Rote. Binary trees with nodes having 0, 1, and 2 children. Séminaire Lotharingien de Combinatoire. (1997)

New Worst-Case Formula Measure

That's not by chance! Using a result about integer sequences [3] we deduced:

1. Every matrix M in $\mathcal{M}$ has exactly **n distinct rotations**. $n = \#$ of columns
2. The **rotation of M that is invertible** exists and is unique.

We observe $|\mathcal{M}| = \prod_{c \in \Sigma} \binom{n}{n_c}$ $\qquad n_c =$ number of ones at the c-th row

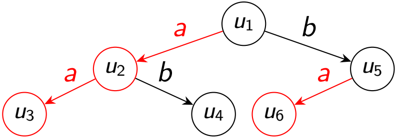
Consequently, $|\mathcal{U}| = \frac{1}{n} \prod_{c \in \Sigma} \binom{n}{n_c}$ and $\mathcal{H}^{wc}(\mathcal{U}) = \sum_{c \in \Sigma} \log \binom{n}{n_c} - \log n$.

3. G. Rote. Binary trees with nodes having 0, 1, and 2 children. Séminaire Lotharingien de Combinatoire. (1997)

Formula Empirical Entropy for Tries

For $w \in \Sigma^k$ and $c \in \Sigma$, consider the integers $\mathbf{n_w}$ and $\mathbf{n_{w,c}}$:

- $\mathbf{n_w} = |\{u \in V \mid u \text{ has context } w\}|$
- $\mathbf{n_{w,c}} = |\{u \in V \mid u \text{ has context } w \text{ and there exists } u \xrightarrow{c} v\}|$



Example: In figure, $\mathbf{n_a} = \mathbf{3}$.

Indeed, u_2, u_3, and u_6 are reached by the string a.

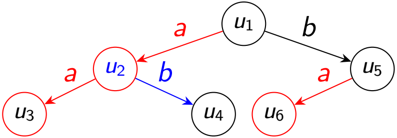
Definition: k-th order empirical entropy $\mathcal{H}_k(\mathcal{T})$

$$\mathcal{H}_k(\mathcal{T}) = \sum_{c \in \Sigma} \sum_{w \in \Sigma^k} \frac{n_{w,c}}{n} \log\left(\frac{n_w}{n_{w,c}}\right) + \frac{n_w - n_{w,c}}{n} \log\left(\frac{n_w}{n_w - n_{w,c}}\right)$$

Formula Empirical Entropy for Tries

For $w \in \Sigma^k$ and $c \in \Sigma$, consider the integers $\mathbf{n_w}$ and $\mathbf{n_{w,c}}$:

- $\mathbf{n_w} = |\{u \in V \mid u \text{ has context } w\}|$
- $\mathbf{n_{w,c}} = |\{u \in V \mid u \text{ has context } w \text{ and there exists } u \xrightarrow{c} v\}|$



Example: In figure, $\mathbf{n_{a,b}} = \mathbf{1}$.

Among the nodes reached by a, only u_2 has an outgoing edge labeled by b.

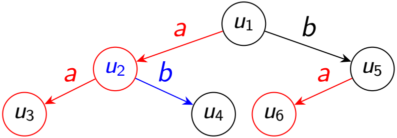
Definition: k-th order empirical entropy $\mathcal{H}_k(\mathcal{T})$

$$\mathcal{H}_k(\mathcal{T}) = \sum_{c \in \Sigma} \sum_{w \in \Sigma^k} \frac{n_{w,c}}{n} \log\left(\frac{n_w}{n_{w,c}}\right) + \frac{n_w - n_{w,c}}{n} \log\left(\frac{n_w}{n_w - n_{w,c}}\right)$$

Formula Empirical Entropy for Tries

For $w \in \Sigma^k$ and $c \in \Sigma$, consider the integers $\mathbf{n_w}$ and $\mathbf{n_{w,c}}$:

- $\mathbf{n_w} = |\{u \in V \mid u \text{ has context } w\}|$
- $\mathbf{n_{w,c}} = |\{u \in V \mid u \text{ has context } w \text{ and there exists } u \xrightarrow{c} v\}|$



Example: In figure, $\mathbf{n_{a,b}} = \mathbf{1}$.

Among the nodes reached by a, only u_2 has an outgoing edge labeled by b.

Definition: k-th order empirical entropy $\mathcal{H}_k(\mathcal{T})$

$$\mathcal{H}_k(\mathcal{T}) = \sum_{c\in\Sigma} \sum_{w\in\Sigma^k} \frac{n_{w,c}}{n} \log\left(\frac{n_w}{n_{w,c}}\right) + \frac{n_w - n_{w,c}}{n} \log\left(\frac{n_w}{n_w - n_{w,c}}\right)$$

Properties for our Entropy Measures

Properties analogous to the string entropies:

1. $n\mathcal{H}_0(\mathcal{T}) = \mathcal{H}^{wc}(\mathcal{T}) + O(\sigma \log n)$
2. $\mathcal{H}_{k+1}(\mathcal{T}) \leq \mathcal{H}_k(\mathcal{T})$, for every $k \geq 0$

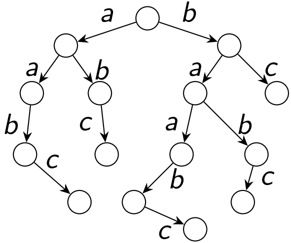
- Worst-case entropy without character frequencies [1] (**not ours!**): $\log \frac{1}{n}\binom{n\sigma}{n-1} = \log \frac{1}{15}\binom{45}{14} \approx$ **33.37 bits**.
- 1st-order empirical entropy (**ours!**) $n\mathcal{H}_1(\mathcal{T}) \approx$ **7.29 bits**

1. *R. Graham, D. Knuth, and O. Patashnik: Concrete Mathematics. Addison-Wesley. (1994)*

Properties for our Entropy Measures

Properties analogous to the string entropies:

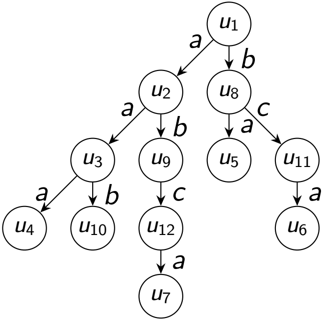
1. $n\mathcal{H}_0(\mathcal{T}) = \mathcal{H}^{wc}(\mathcal{T}) + O(\sigma \log n)$
2. $\mathcal{H}_{k+1}(\mathcal{T}) \leq \mathcal{H}_k(\mathcal{T})$, for every $k \geq 0$



- Worst-case entropy without character frequencies [1] (**not ours!**): $\log \frac{1}{n}\binom{n\sigma}{n-1} = \log \frac{1}{15}\binom{45}{14} \approx$ **33.37 bits**.
- 1st-order empirical entropy (**ours!**) $n\mathcal{H}_1(\mathcal{T}) \approx$ **7.29 bits**

1. *R. Graham, D. Knuth, and O. Patashnik: Concrete Mathematics. Addison-Wesley. (1994)*

XBWT of a trie



$out(u) \leftarrow$ set of outgoing labels of u

$u_1, u_2, \ldots, u_n \leftarrow$ nodes sorted **co-lexicographically**

Definition: XBWT [4]

$$\text{XBWT}(\mathcal{T}) = out(u_1), out(u_2), \ldots, out(u_n)$$

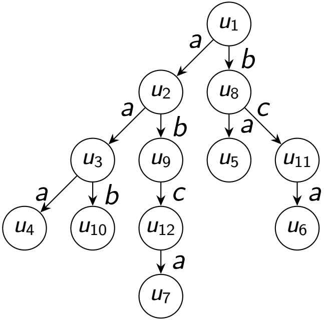
We can **compress** and **index** (count queries) a trie in:

$$n\mathcal{H}_k(\mathcal{T}) + o(n) \quad \forall k = o(\log_\sigma n)$$

co-lex	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}
XBWT	a b	a b	a b					a c	c		a	a

4. P. Ferragina et al. Compressing and Indexing Labeled Trees, with Applications. J. ACM. (2009)

XBWT of a trie



$out(u) \leftarrow$ set of outgoing labels of u

$u_1, u_2, \ldots, u_n \leftarrow$ nodes sorted **co-lexicographically**

Definition: XBWT [4]

$$\text{XBWT}(\mathcal{T}) = out(u_1), out(u_2), \ldots, out(u_n)$$

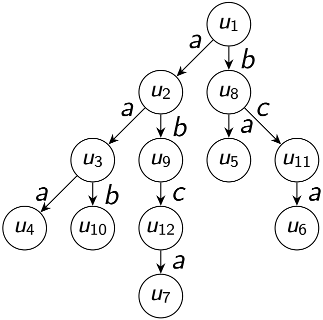
We can **compress** and **index** (count queries) a trie in:

$$n\mathcal{H}_k(\mathcal{T}) + o(n) \quad \forall k = o(\log_\sigma n)$$

co-lex	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}
XBWT	a b	a b	a b					a c	c		a	a

4. P. Ferragina et al. Compressing and Indexing Labeled Trees, with Applications. J. ACM. (2009)

XBWT runs



XBWT run-break if: $c \in out(u_i)$ and $c \notin out(u_{i+1})$

r-index for tries in: $O(r \log n) + o(n)$ bits [5]

We proved $r \leq n\mathcal{H}_k(\mathcal{T}) + \sigma^{k+1}$

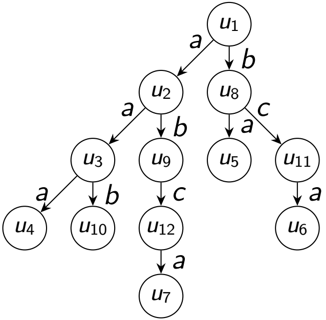
(similar relation for strings! [6])

co-lex	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}
XBWT	a b	a b	a b					a c	 c		a	a

5. N. Prezza. On Locating Paths in Compressed Tries. SODA. (2021)

6. V. Mäkinen, G. Navarro. Succinct Suffix Arrays Based on RLE. Nordic Journal of Computing. (2005)

XBWT runs



XBWT run-break if: $c \in out(u_i)$ and $c \notin out(u_{i+1})$

r-index for tries in: $O(r \log n) + o(n)$ bits [5]

We proved $r \leq n\mathcal{H}_k(\mathcal{T}) + \sigma^{k+1}$

(similar relation for strings! [6])

co-lex	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}
XBWT	a b	a b	a b					a c	c		a	a

5. N. Prezza. On Locating Paths in Compressed Tries. SODA. (2021)

6. V. Mäkinen, G. Navarro. Succinct Suffix Arrays Based on RLE. Nordic Journal of Computing. (2005)

Thank you for your attention ☺

1. Provide an **alternative proof** for the formula $|\mathcal{U}| = \frac{1}{n} \prod_{c \in \Sigma} \binom{n}{n_c}$
2. Introducing corresponding **worst-case entropy** $\mathcal{H}^{wc}(\mathcal{U})$
3. Introducing an **empirical entropy for tries** $\mathcal{H}_k(\mathcal{T})$
4. **Compress** and **index** a trie in $n\mathcal{H}_k(\mathcal{T}) + o(n)$ bits using the **XBWT**