A Comparison of Empirical Tree Entropies

Danny Hucke, Markus Lohrey, and Louisa Seelbach Benkner

University of Siegen, Germany, {hucke,lohrey,seelbach}@eti.uni-siegen.de

Abstract. Whereas for strings, higher-order empirical entropy is the standard entropy measure, several different notions of empirical entropy for trees have been proposed in the past, notably label entropy, degree entropy, conditional versions of the latter two, and empirical entropy of trees (here, called label-shape entropy). In this paper, we carry out a systematic comparison of these entropy measures. We underpin our theoretical investigations by experimental results with real XML data.

1 Introduction

In the area of string compression the notion of higher order empirical entropy yields a well established measure for the compressibility of a string. Roughly speaking, the k^{th}-order empirical entropy of a string is the expected uncertainty about the symbol at a certain position, given the k-preceding symbols. In fact, except for some modifications (as the k^{th}-order modified empirical entropy from [19]) the authors are not aware of any other empirical entropy measure for strings ("empirical" refers to the fact that the entropy is defined for the string itself and not a certain probability distribution on strings). For many string compressors, worst-case bounds on the length of a compressed string in terms of the k^{th}-order empirical entropy are known [11,19,20]. For further aspects of higher-order empirical entropy see [8].

If one goes from strings to trees the situation becomes different. Let us first mention that the area of tree compression (and compression of structured data in general) is currently a very active area, which is motivated by the appearance of large tree data in applications like XML processing. Common tree compression techniques are based on succinct tree encodings [5,6,12,17,21], grammar-based tree compressors [13,9,14,18], directed acyclic graphs [7,3] and top dags [2,1]. In recent years, several notions of empirical tree entropy have been proposed with the aim of quantifying the compressibility of a given tree. Let us briefly discuss these entropies in the following paragraphs (all entropies below are unnormalized; the corresponding normalized entropies are obtained by dividing by the tree size).

Ferragina et al. [5,6] introduced the k^{th}-order label entropy $H_k^\ell(t)$ of a node-labeled unranked[1] tree t. Its normalized version is the expected uncertainty about the label of a node v, given the so-called *k-label-history* of v, which contains

[1] Unranked means that there is no bound on the number of children of a node. Moreover, we only consider ordered trees, where the children of a node are linearly ordered.

the k first labels on the path from v's parent node to the root. The k^{th}-order label entropy is not useful for unlabeled trees since it ignores the tree shape.

In [17], Jansson et al. introduce the *degree entropy* $H^{\deg}(t)$, which is the (unnormalized) 0^{th}-order empirical entropy of the node degrees occurring in the unranked tree t. Degree entropy is mainly made for unlabeled trees since it ignores node labels, but in combination with label entropy it yields a reasonable measure for the compressibility of a tree: every node-labeled unranked tree of size n in which σ many different node labels occur can be stored in $H_k^{\ell}(t) + H^{\deg}(t) + o(n + n \log \sigma)$ bits if σ is not too big; see Theorem 2. Note that the (unnormalized) degree entropy of a binary tree with n leaves converges to $2n - o(n)$ since a binary tree with n leaves has exactly $n - 1$ nodes of degree 2.

Recently, Ganczorz [12] defined relativized versions of k^{th}-order label entropy and degree entropy: the k^{th}-order degree-label entropy $H_k^{\deg,\ell}(t)$ and the k^{th}-order label-degree entropy $H_k^{\ell,\deg}(t)$. The normalized version of $H_k^{\deg,\ell}(t)$ is the expected uncertainty about the label of a node v of t, given (i) the k-label-history of v and (ii) the degree of v, whereas the normalized version of $H_k^{\ell,\deg}(t)$ is the expected uncertainty about the degree of a node v, given (i) the k-label-history of v and (ii) the label of v. Ganczorz [12] proved that every node-labeled unranked tree of size n can be stored in $H_k^{\ell}(t) + H_k^{\ell,\deg}(t) + o(n + n \log \sigma)$ bits as well as in $H^{\deg}(t) + H_k^{\deg,\ell}(t) + o(n + n \log \sigma)$ bits (again assuming σ is not too big); see Theorem 2. Note that for unlabeled trees t, we have $H_k^{\ell}(t) + H_k^{\ell,\deg}(t) = H^{\deg}(t) + H_k^{\deg,\ell}(t) = H^{\deg}(t)$, which for unlabeled binary trees is equal to the information theoretic upper bound $2n - o(n)$ (with n the number of leaves).

Motivated by the inability of the existing entropies for measuring the compressibility of unlabeled binary trees, we introduced in [14] a new entropy for binary trees (possibly with labels) that we called k^{th}-order empirical entropy $H_k(t)$. In order to distinguish it better from the existing tree entropies we prefer the term *k^{th}-order label-shape entropy* in this paper. The main idea is to extend k-label-histories in a binary tree by adding to the labels of the k predecessors of a node v also the k last directions (0 for left, 1 for right) on the path from the root to v. We call this extended label history simply the k-history of v. The normalized version of $H_k(t)$ is the expected uncertainty about the pair consisting of the label of a node and the information whether it is a leaf or an internal node, given the k-history of the node. The main result of [14] states that a node-labeled binary tree t can be stored in $H_k(t) + o(n + n \log \sigma)$ bits using a grammar-based code building on tree straight-line programs. We also defined in [14] the k^{th}-order label-shape entropy of an unranked node-labeled tree t by taking the k^{th}-order label-shape entropy of the first-child next-sibling encoding of t.

The goal of this paper is to compare the entropy variants $H_k^{\ell}(t) + H^{\deg}(t)$, $H_k^{\ell}(t) + H_k^{\ell,\deg}(t)$, $H^{\deg}(t) + H_k^{\deg,\ell}(t)$, and $H_k(t)$. Our results for unranked node-labeled trees are summarized in Figure 1. Let us explain the meaning of the arrows in Figure 1: For two entropy notions H and H', an arrow $H \xrightarrow{\exists\, o} H'$ means that there is a sequence of unranked node-labeled trees t_n ($n \geq 1$) such

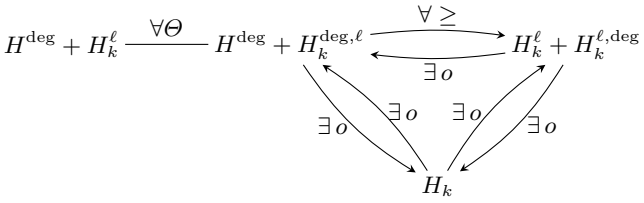

Fig. 1. Comparison of the entropy notions for unranked node-labeled trees. The meaning of the red and green arrows is explained in the main text.

that (i) the function $n \mapsto |t_n|$ is strictly increasing and (ii) $H(t_n) \leq o(H'(t_n))$ (in most cases we prove an exponential separation). The meaning of the arrow with label $\forall \geq$ is that $H^{\deg}(t)+H_k^{\deg,\ell}(t) \geq H_k^{\ell}(t)+H_k^{\ell,\deg}(t)$ for every unranked node-labeled tree t, whereas the edge with label $\forall\Theta$ means that $H^{\deg}(t) + H_k^{\deg,\ell}(t)$ and $H^{\deg}(t) + H_k^{\ell}(t)$ are equivalent up to fixed multiplicative constants (which are 1 and 2).

We also investigate the relationship between the entropies for node-labeled binary trees and unranked unlabeled trees (the case of unlabeled binary trees is not really interesting as explained above). An unranked unlabeled tree t of size n can be represented with $H^{\deg}(t) + o(n)$ bits [17]. Here, we prove that $H_k(t) \leq 2H^{\deg}(t) + 2\log_2(n) + 4$ for every unranked unlabeled tree t.

Finally, we underpin our theoretical results by experimental results with real XML data from XMLCompBench (`http://xmlcompbench.sourceforge.net`). For each XML document we consider the corresponding tree structure t (obtained by removing all text values and attributes) and compute $H_k^{\ell}(t)+H^{\deg}(t)$, $H_k^{\ell}(t)+H_k^{\ell,\deg}(t)$, $H^{\deg}(t)+H_k^{\deg,\ell}(t)$, and $H_k(t)$. The results are summarized in Table 1 on page 14. Our experiments indicate that the upper bound on the number of bits needed by the compressed data structure in [14] is the strongest for real XML data since the k^{th}-order label-shape entropy (for $k > 0$) is significantly smaller than all other entropy values for all XMLs that we have examined.

Let us remark that Ganczorz's succinct tree representations [12] that achieve (up to low-order terms) the entropies $H_k^{\ell}(t) + H_k^{\ell,\deg}(t)$ and $H^{\deg}(t) + H_k^{\deg,\ell}(t)$, respectively, allow constant query times for a large number of tree queries. For the entropy $H_k(t)$ such a result is not known. The tree representation from [14] is based on tree straight-line programs, which can be queried in logarithmic time (if we assume logarithmic height of the grammar, which can be enforced by [10]).

Missing proofs can be found in the long version [16].

2 Preliminaries

With $\mathbb{N}$ we denote the natural numbers including 0. Let $w = a_1 \cdots a_l \in \Gamma^*$ be a word over an alphabet Γ. With $|w| = l$ we denote the length of w. Let ε denote the empty word. We use the standard $\mathcal{O}$-notation. If $b > 1$ is a constant, then we write $\mathcal{O}(\log n)$ for $\mathcal{O}(\log_b n)$. Moreover, terms $\log_b n$ with $b \geq 1$ are implicitly

replaced by $\log_{b'} n$ for $b' = \max\{2, b\}$. We make the convention that $0 \cdot \log(0) = 0$ and $0 \cdot \log(x/0) = 0$ for $x \geq 0$. The well-known log-sum inequality (see e.g. [4, Theorem 2.7.1]) states:

Lemma 1 (Log-Sum inequality). *Let $a_1, a_2, \ldots, a_l, b_1, b_2, \ldots, b_l \geq 0$ be real numbers. Moreover, let $a = \sum_{i=1}^{l} a_i$ and $b = \sum_{i=1}^{l} b_i$. Then*

$$a \log_2 \left(\frac{b}{a}\right) \geq \sum_{i=1}^{l} a_i \log_2 \left(\frac{b_i}{a_i}\right).$$

2.1 Unranked trees

Let Σ denote a finite alphabet of size $|\Sigma| = \sigma \geq 1$. Later, we need a fixed, distinguished symbol from Σ that we denote with $\Box \in \Sigma$. We consider *Σ-labeled unranked ordered trees*, where "Σ-labeled" means that every node is labeled by a symbol from the alphabet Σ, "ordered" means that the children of a node are totally ordered, and "unranked" means that the number of children of a node (also called its *degree*) can be any natural number. In particular, the degree of a node does not depend on the node's label or vice versa. Let us denote by $\mathcal{T}(\Sigma)$ the set of all such trees. Formally, the set $\mathcal{T}(\Sigma)$ is inductively defined as the smallest set of expressions such that if $a \in \Sigma$ and $t_1, \ldots, t_n \in \mathcal{T}(\Sigma)$ then also $a(t_1 \cdots t_n) \in \mathcal{T}(\Sigma)$. This expression represents a tree with an a-labeled root whose direct subtrees are $t_1, \ldots, t_n$. Note that for the case $n = 0$ we obtain the tree $a()$, for which we also write a. The *size* $|t|$ of $t \in \mathcal{T}(\Sigma)$ is the number of occurrences of labels from Σ in t, i.e., $a(t_1 \cdots t_n) = 1 + \sum_{i=1}^{n} |t_i|$. We identify an unranked tree with a graph in the usual way, where each node is labeled with a symbol from Σ. Let $V(t)$ denote the set of nodes of a tree $t \in \mathcal{T}(\Sigma)$. We have $|V(t)| = |t|$. The label of a node $v \in V(t)$ is denoted with $\ell(v) \in \Sigma$. Moreover, we write $\deg(v) \in \mathbb{N}$ for the degree of v. An important special case of unranked trees are *unlabeled unranked trees*: They can be considered as labeled unranked trees over a singleton alphabet $\Sigma = \{a\}$.

For a node $v \in V(t)$ of a tree t, we define its *label-history* $h^\ell(v) \in \Sigma^*$ inductively: for the root node v_0, we set $h^\ell(v_0) = \varepsilon$ and for a child node w of a node v of t, we set $h^\ell(w) = h^\ell(v)\,\ell(v)$. In other words: $h^\ell(v)$ is obtained by concatenating the node labels along the unique path from the root to v. The label of v is not part of the label-history of v. The *k-label-history* $h^\ell_k(v)$ of a tree node $v \in V(t)$ is defined as the length-k suffix of $\Box^k h^\ell(v)$, where $\Box$ is a fixed dummy symbol in Σ. This means that if the depth of v in t is greater than k, then $h^\ell_k(v)$ lists the last k node labels along the path from the root to node v. If the depth of v in t is at most v, then we pad its label-history $h^\ell(v)$ with the symbol $\Box$ such that $h^\ell_k(v) \in \Sigma^k$. For $z \in \Sigma^k$, $a \in \Sigma$ and $i \in \mathbb{N}$ we set

$$n^t_z = |\{v \in V(t) \mid h^\ell_k(v) = z\}|, \tag{1}$$

$$n^t_i = |\{v \in V(t) \mid \deg(v) = i\}|, \tag{2}$$

$$n^t_{z,i} = |\{v \in V(t) \mid h^\ell_k(v) = z \text{ and } \deg(v) = i\}|, \tag{3}$$

$$n^t_{z,a} = |\{v \in V(t) \mid h^\ell_k(v) = z \text{ and } \ell(v) = a\}|, \tag{4}$$

$$n^t_{z,i,a} = |\{v \in V(t) \mid h^\ell_k(v) = z,\ \ell(v) = a \text{ and } \deg(v) = i\}|. \tag{5}$$

In order to avoid ambiguities in these notations we should assume that $\Sigma \cap \mathbb{N} = \emptyset$. Moreover, when writing $n^t_{z,i}$ (resp., $n^t_{z,a}$) then, implicitly, i (resp., a) always belongs to $\mathbb{N}$ (resp., Σ).

2.2 Binary trees

An important subset of $\mathcal{T}(\Sigma)$ is the set $\mathcal{B}(\Sigma)$ of *labeled binary trees* over the alphabet Σ. A binary tree is a tree in $\mathcal{T}(\Sigma)$, where every node has either exactly two children or is a leaf. Formally, $\mathcal{B}(\Sigma)$ is inductively defined as the smallest set of terms over Σ such that (i) $\Sigma \subseteq \mathcal{B}(\Sigma)$ and (ii) if $t_1, t_2 \in \mathcal{B}(\Sigma)$ and $a \in \Sigma$, then $a(t_1 t_2) \in \mathcal{B}(\Sigma)$. An *unlabeled binary tree* can be considered as a binary tree over the singleton alphabet $\Sigma = \{a\}$. The *first-child next-sibling encoding* (or shortly *fcns-encoding*) transforms a tree $t \in \mathcal{T}(\Sigma)$ into a binary tree $t \in \mathcal{B}(\Sigma)$. We define it more generally for an ordered sequence of unranked trees $s = t_1 t_2 \cdots t_n$ (a so-called forest) inductively as follows (recall that $\Box \in \Sigma$ is a fixed distinguished symbol in Σ): $\mathrm{fcns}(s) = \Box$ for $n = 0$ and if $n \geq 1$ and $t_1 = a(t'_1 \cdots t'_m)$ then $\mathrm{fcns}(s) = a(\mathrm{fcns}(t'_1 \cdots t'_m)\, \mathrm{fcns}(t_2 \cdots t_n))$. Thus, the left (resp. right) child of a node in $\mathrm{fcns}(s)$ is the first child (resp., right sibling) of the node in s or a $\Box$-labeled leaf, if it does not exist.

For the special case of binary trees, we extend the label history of a node to its full history, which we just call its history. Intuitively, the history of a node v records all information that can be obtained by walking from the root of the tree straight down to the node v. In addition to the node labels this also includes the directions (left/right) of the descending edges. For an integer $k \geq 0$ let

$$\mathcal{L}_k = (\Sigma\{0,1\})^k = \{a_1 i_1 a_2 i_2 \cdots a_k i_k \mid a_j \in \Sigma, i_j \in \{0,1\} \text{ for } 1 \leq j \leq k\}.$$

For a node v of a binary tree t, we define its *history* $h(v) \in (\Sigma\{0,1\})^*$ inductively as follows: For the root node v_0, we set $h(v_0) = \varepsilon$. For a left child node w of a node v of t, we set $h(w) = h(v)\ell(v)0$ and for a right child node w of v, we set $h(w) = h(v)\ell(v)1$ (recall that $\ell(v)$ is the label of v). That is, in order to obtain $h(v)$, while descending in the tree from the root node to the node v, we alternately write down the current node label from Σ and the direction into which we descend (0 if we descend to a left child, 1 if we descend to a right child). Note that the symbol that labels v is not part of the history $h(v)$. The *k-history* of a node v is then defined as the length-$2k$ suffix of the word $(\Box 0)^k h(v)$, where $\Box$ is again a fixed dummy symbol in Σ. This means that if the depth of v in t is greater than k, then $h_k(v)$ describes the last k directions and node labels along the path from the root to node v. If the depth of v in t is at most k, then we pad the history of v with $\Box$'s and zeroes such that $h_k(v) \in \mathcal{L}_k$. For a node v of a binary tree we define $\lambda(v) = (\ell(v), \deg(v)) \in \Sigma \times \{0,2\}$. For $z \in \mathcal{L}_k$ and $\tilde{a} \in \Sigma \times \{0,2\}$, we finally define

$$m^t_z = |\{v \in V(t) \mid h_k(v) = z\}|, \tag{6}$$

$$m^t_{z,\tilde{a}} = |\{v \in V(t) \mid h_k(v) = z \text{ and } \lambda(v) = \tilde{a}\}|. \tag{7}$$

3 Empirical entropy for trees

In this section we formally define the various entropy measures that were mentioned in the introduction. Note that in all cases we define so-called unnormalized entropies, which has the advantage that we do not have to multiply with the size of the tree in bounds for the encoding size of a tree. Note that in [5,6,12,17] the authors define normalized entropies. In each case, one obtains the normalized entropy by dividing the corresponding unnormalized entropy by the tree size.

Label entropy. The first notion of empirical entropy for trees was introduced in [5]. In order to distinguish notions, we call the entropy from [5] *label entropy*. It is defined for unranked labeled trees $t \in \mathcal{T}(\Sigma)$: the k^{th}-order *label entropy* $H^\ell_k(t)$ of t is defined as follows, where n^t_z and $n^t_{z,a}$ are from (1) and (4), respectively:

$$H^\ell_k(t) = \sum_{z \in \Sigma^k} \sum_{a \in \Sigma} n^t_{z,a} \log_2 \left(\frac{n^t_z}{n^t_{z,a}} \right). \tag{8}$$

We remark that in [5], it is not explicitly specified how to deal with nodes, whose label-history is shorter than k. There are three natural variants: (i) padding label-histories with a symbol $\square \in \Sigma$ (this is our choice), (ii) padding label-histories with a fresh symbol $\diamond \notin \Sigma$, or equivalently, allowing label-histories of length smaller than k, and (iii) ignoring nodes whose label-history is shorter than k. However, similar considerations as in the appendix of [15] show that these approaches yield the same k^{th}-order label entropy up to an additional additive term of at most $m^<(1 + 1/\ln(2) + \log_2(\sigma|t|/m^<))$, where $m^<$ is the number of nodes at depth less than k in t.

Degree entropy. Another notion of empirical entropy for trees is the entropy measure from [17], which we call *degree entropy*. Degree entropy is primarily made for unlabeled unranked trees, as it completely ignores node labels. Nevertheless the definition works for trees $t \in \mathcal{T}(\Sigma)$ over any alphabet Σ. For a tree $t \in \mathcal{T}(\Sigma)$, the degree entropy $H^{\deg}(t)$ is the 0^{th}-order entropy of the node degrees (n^t_i is from (2)):

$$H^{\deg}(t) = \sum_{i=0}^{|t|} n^t_i \log_2 \left(\frac{|t|}{n^t_i} \right).$$

For the special case of unlabeled trees the following result was shown in [17]:

Theorem 1 ([17, Theorem 1]). *Let t be an unlabeled unranked tree. Then t can be represented with $H^{\deg}(t) + \mathcal{O}(|t| \log\log(|t|)/\log|t|)$ bits.*

Label-degree entropy and degree-label entropy. Recently, two combinations of the label entropy from [5] and the degree entropy from [17] were proposed in [12]. We call these two entropy measures *label-degree entropy* and *degree-label entropy*. Both notions are defined for unranked node-labeled trees. Let $t \in \mathcal{T}(\Sigma)$ be such a tree. The k^{th}-order *label-degree entropy* $H_k^{\ell,\deg}(t)$ of t from [12] is defined as follows, where $n^t_{z,a}$ and $n^t_{z,i,a}$ are from (4) and (5), respectively:

$$H_k^{\ell,\deg}(t) = \sum_{z\in\Sigma^k} \sum_{a\in\Sigma} \sum_{i=0}^{|t|} n^t_{z,i,a} \log_2 \left(\frac{n^t_{z,a}}{n^t_{z,i,a}} \right).$$

The k^{th}-order *degree-label entropy* $H_k^{\deg,\ell}(t)$ of t from [12] is defined as follows, where $n^t_{z,i}$ and $n^t_{z,i,a}$ are from (3) and (5), respectively:

$$H_k^{\deg,\ell}(t) = \sum_{z\in\Sigma^k} \sum_{i=0}^{|t|} \sum_{a\in\Sigma} n^t_{z,i,a} \log_2 \left(\frac{n^t_{z,i}}{n^t_{z,i,a}} \right).$$

In order to deal with nodes whose label-history is shorter than k one can again choose one of the three alternatives (i)–(iii) that were mentioned after (8). In [12], variant (ii) is chosen, while the above definitions correspond to choice (i). However, as for the label entropy one can show that these variants only differ by a small additive term of at most $m^<(1/\ln(2) + \log_2(\sigma|t|/m^<))$ in the case of the degree-label entropy, respectively, $m^<(1/\ln(2) + \log_2|t|)$ in the case of the label-degree entropy, where $m^<$ is the number of nodes at depth less than k.

By [12], the following inequalities hold:

Lemma 2. *For every $t \in \mathcal{T}(\Sigma)$, $H_k^{\ell,\deg}(t) \le H^{\deg}(t)$ and $H_k^{\deg,\ell}(t) \le H_k^{\ell}(t)$.*

Moreover, one of the main results of [12] states the following bounds:

Theorem 2 ([12, Theorem 12]). *Let $t \in \mathcal{T}(\Sigma)$, with $\sigma \le |t|^{1-\alpha}$ for some $\alpha > 0$. Then t can be represented in*

$$H + \mathcal{O}\left(\frac{|t|k\log\sigma + |t|\log\log_\sigma|t|}{\log_\sigma|t|} \right),$$

bits, where H is one of $H^{\deg}(t)+H_k^{\ell}(t)$, $H_k^{\ell}(t)+H_k^{\ell,\deg}(t)$, or $H^{\deg}(t)+H_k^{\deg,\ell}(t)$.

Label-shape entropy. Another notion of empirical entropy for trees which incorporates both node labels and tree structure was recently introduced in [14]: Let us start with a binary tree $t \in \mathcal{B}(\Sigma)$. The k^{th}-order label-shape entropy $H_k(t)$ of t (in [14] it is simply called the k^{th}-order empirical entropy of t) is

$$H_k(t) = \sum_{z\in\mathcal{L}_k} \sum_{\tilde{a}\in\Sigma\times\{0,2\}} m^t_{z,\tilde{a}} \log_2 \left(\frac{m^t_z}{m^t_{z,\tilde{a}}} \right), \tag{9}$$

where m_z^t and $m_{z,\tilde{a}}^t$ are from (6) and (7), respectively. Now let $t \in \mathcal{T}(\Sigma)$ be an unranked tree and recall that $\mathrm{fcns}(t) \in \mathcal{B}(\Sigma)$. The k^{th}-order label-shape entropy $H_k(t)$ of t is defined as

$$H_k(t) = H_k(\mathrm{fcns}(t)). \tag{10}$$

The following result is shown in [14] using a grammar-based encoding of trees:

Theorem 3. *Every tree $t \in \mathcal{T}(\Sigma)$ can be represented within the following bound (in bits):*

$$H_k(t) + \mathcal{O}\left(\frac{k|t|\log\sigma}{\log_\sigma|t|}\right) + \mathcal{O}\left(\frac{|t|\log\log_\sigma|t|}{\log_\sigma|t|}\right) + \sigma.$$

Note that for binary trees, there are two possibilities how to compute the label-shape entropy $H_k(t)$. The first is to compute the label-shape entropy as defined in (9), the second is to consider the binary tree as an unranked tree and compute the label-shape entropy of its first-child next-sibling encoding as defined in (10). The following lemma from [15] states that if we consider the first-child next-sibling encoding of the binary tree instead of the binary tree itself, the k^{th}-order label-shape entropy does not increase if we double the value of k:

Lemma 3. *Let $t \in \mathcal{B}(\Sigma)$ be a binary tree with first-child next-sibling encoding $fcns(t) \in \mathcal{B}(\Sigma)$. Then $H_{2k}(fcns(t)) \leq H_{k-1}(t)$ for $1 \leq k \leq n$.*

In contrast to Lemma 3, there are families of binary trees t_n where $H_k(t_n) \in \Theta(n-k)$ and $H_k(\mathrm{fcns}(t_n)) \in \Theta(\log(n-k))$ [15].

4 Comparison of the empirical entropy notions

As we have seen in Theorems 2 and 3, entropy bounds for the number of bits needed to represent an unranked labeled tree t are achievable by $H_k(t)$, $H_k^\ell(t) + H_k^{\ell,\deg}(t)$, $H^{\deg}(t) + H_k^{\deg,\ell}(t)$, and $H^{\deg}(t) + H_k^\ell(t)$, where in all cases we have to add a low-order term. The term $H^{\deg}(t) + H_k^\ell(t)$ is lower-bounded by $H_k^\ell(t) + H_k^{\ell,\deg}(t)$ and $H^{\deg}(t) + H_k^{\deg,\ell}(t)$ by Lemma 2. For the special case of unlabeled unranked trees, $H^{\deg}(t)$ (plus low-order terms) is an upper bound on the encoding length (see Theorem 1) as well. Let us also remark that $H_{k'}(t) \leq H_k(t)$ for $k < k'$ and analogously for H_k^ℓ, $H_k^{\ell,\deg}$, and $H_k^{\deg,\ell}$.

4.1 Binary trees

Let us start with unlabeled binary trees, i.e., trees $t \in \mathcal{B}(\{a\})$ over the unary alphabet $\Sigma = \{a\}$. As $\Sigma = \{a\}$, the fixed dummy symbol used to pad k-histories and k-label-histories is $\square = a$. The following lemma follows from the fact that every binary tree of size $2n-1$ consists of n nodes of degree 0 and $n-1$ nodes of degree 2:

Lemma 4. *Let t be an unlabeled binary tree with n leaves and thus $|t| = 2n-1$. Then $H^{\deg}(t) = H_k^{\ell,\deg}(t) = (2-o(1))n$.*

For the following lower bound one can take for t_n a left-degenerate chain of height n (formally: $t_1 = a$ and $t_n = a(t_{n-1}\, a)$ for $n \geq 2$).

Lemma 5. *There exists a family of unlabeled binary trees $(t_n)_{n\geq 1}$ such that $|t_n| = 2n-1$ and $H_k(t_n) \leq \log_2(en)$ for all $n \geq 1$ and $1 \leq k \leq n$.*

Lemmas 4 and 5 already indicate that all entropies considered in this paper except for the label-shape entropy are not interesting for unlabeled binary trees. For every unlabeled binary tree t with n leaves (and $2n-1$ nodes) we have: $H_k^{\ell}(t) = H_k^{\deg,\ell} = 0$, as every node of t has the same label, and $H_k^{\ell}(t)+H_k^{\ell,\deg}(t) = H^{\deg}(t) + H_k^{\deg,\ell}(t) = H_k^{\ell}(t) + H^{\deg}(t) = H^{\deg}(t)$, and these values are lower bounded by $2n(1-o(1))$ (Lemma 4). In contrast, the label-shape entropy (9) is able to capture regularities in unlabeled binary trees (and attains different values for different binary trees of the same size).

Let us now look at binary trees $t \in \mathcal{B}(\Sigma)$, where Σ is arbitrary. As in the special case of unlabeled binary trees, we find that $H^{\deg}(t) = 2n(1-o(1))$ for every binary tree t of size $2n-1$ (the node labels do not influence $H^{\deg}(t)$), which implies $H^{\deg}(t) + H_k^{\deg,\ell}(t) \geq 2n(1-o(1))$. The following lemma shows that $H_k(t)$ is always bounded by $H_k^{\ell}(t) + H_k^{\ell,\deg}(t)$ and $H^{\deg}(t) + H_k^{\deg,\ell}(t)$ (and hence also $H_k^{\ell}(t) + H^{\deg}(t)$) for $t \in \mathcal{B}(\Sigma)$.

Lemma 6. *Let $t \in \mathcal{B}(\Sigma)$ be a binary tree. Then (i) $H_k(t) \leq H_k^{\ell}(t) + H_k^{\ell,\deg}(t)$ and (ii) $H_k(t) \leq H^{\deg}(t) + H_k^{\deg,\ell}(t)$.*

Proof. We start with proving statement (i): We have

$$\begin{aligned}
H_k(t) &= \sum_{z\in\mathcal{L}_k}\sum_{a\in\Sigma}\sum_{i\in\{0,2\}} m^t_{z,(a,i)} \log_2\left(\frac{m^t_z}{m^t_{z,(a,i)}}\right) \\
&= \sum_{z\in\mathcal{L}_k}\sum_{a\in\Sigma}\left(m^t_{z,(a,0)} + m^t_{z,(a,2)}\right)\log_2\left(\frac{m^t_z}{m^t_{z,(a,0)} + m^t_{z,(a,2)}}\right) \\
&+ \sum_{z\in\mathcal{L}_k}\sum_{a\in\Sigma}\sum_{i\in\{0,2\}} m^t_{z,(a,i)} \log_2\left(\frac{m^t_{z,(a,0)} + m^t_{z,(a,2)}}{m^t_{z,(a,i)}}\right) \\
&\leq \sum_{z\in\Sigma^k}\sum_{a\in\Sigma} n^t_{z,a}\log_2\left(\frac{n^t_z}{n^t_{z,a}}\right) + \sum_{z\in\Sigma^k}\sum_{a\in\Sigma}\sum_{i\in\{0,2\}} n^t_{z,i,a}\log_2\left(\frac{n^t_{z,a}}{n^t_{z,i,a}}\right) \\
&= H_k^{\ell}(t) + H_k^{\ell,\deg}(t),
\end{aligned}$$

where the inequality in the second last line follows from the log-sum inequality (Lemma 1) and the last equality follows from the fact that in a binary tree, every node is either of degree 0 or 2. Statement (ii) can be shown in a similar way:

$$H_k(t) = \sum_{z\in\mathcal{L}_k}\sum_{a\in\Sigma}\sum_{i\in\{0,2\}} m^t_{z,(a,i)} \log_2\left(\frac{m^t_z}{m^t_{z,(a,i)}}\right)$$

$$
\begin{aligned}
&= \sum_{z\in\mathcal{L}_k}\sum_{i\in\{0,2\}}\left(\sum_{a\in\Sigma} m^t_{z,(a,i)}\right)\log_2\left(\frac{m^t_z}{\sum_{a\in\Sigma} m^t_{z,(a,i)}}\right)\\
&+ \sum_{z\in\mathcal{L}_k}\sum_{a\in\Sigma}\sum_{i\in\{0,2\}} m^t_{z,(a,i)}\log_2\left(\frac{\sum_{a\in\Sigma} m^t_{z,(a,i)}}{m^t_{z,(a,i)}}\right)\\
&\leq \sum_{i\in\{0,2\}} n^t_i\log_2\left(\frac{|t|}{n^t_i}\right) + \sum_{z\in\Sigma^k}\sum_{a\in\Sigma}\sum_{i\in\{0,2\}} n^t_{z,i,a}\log_2\left(\frac{n^t_{z,i}}{n^t_{z,i,a}}\right)\\
&= H^{\deg}(t) + H^{\deg,\ell}_k(t),
\end{aligned}
$$

where the inequality follows again from the log-sum inequality. □

4.2 Unlabeled unranked trees

In this subsection, we consider unranked trees $t \in \mathcal{T}(\Sigma)$ over the unary alphabet $\Sigma = \{a\}$. As $\Sigma = \{a\}$, the fixed dummy symbol used to pad k-histories and k-label-histories is $\square = a$. Moreover, note that in order to compute $H_k(t)$ for an unranked tree $t \in \mathcal{T}(\Sigma)$, we have to consider fcns(t), which is an unlabeled binary tree (we must take $\square = a$ by our conventions for the dummy symbol; hence the fresh $\square$-labeled leaves in fcns(t) are labeled with a, too). As in the case of unlabeled binary trees, we observe that some entropy measures, in particular those that involve labels, only attain trivial values for unranked unlabeled trees. More precisely, for every tree $t \in \mathcal{T}(\{a\})$ we have $H^{\ell}_k(t) = H^{\deg,\ell}_k(t) = 0$, as every node has the same label a, and $H^{\deg}(t) = H^{\ell,\deg}_k(t)$, as every node has the same k-label-history and the same label. Moreover, we get $H^{\ell}_k(t) + H^{\ell,\deg}_k(t) = H^{\deg}(t) + H^{\deg,\ell}_k(t) = H^{\deg}(t) + H^{\ell}_k(t) = H^{\deg}(t)$. By this observation, we only compare $H_k(t)$ with $H^{\deg}(t)$ for $t \in \mathcal{T}(\{a\})$ in this subsection. By Lemmas 4 and 5, there exists a family of unlabeled trees $(t_n)_{n\geq 1}$ such that $|t_n| = \Theta(n)$ and for which $H_k(t_n)$ is exponentially smaller than $H^{\deg}(t_n)$. For general unranked unlabeled trees, we have the following result; see [16] for the proof.

Theorem 4. *For every unlabeled unranked tree t with $|t| \geq 2$ and integer $k \geq 1$, we have $H_k(t) \leq 2H^{\deg}(t) + 2\log_2(|t|) + 4$.*

As $H^{\deg}(t) = H^{\ell}_k(t) + H^{\ell,\deg}_k(t) = H^{\deg}(t) + H^{\deg,\ell}_k(t)$ for every tree $t \in \mathcal{T}(\{a\})$ and $k \geq 0$, we obtain the following corollary from Theorem 4:

Corollary 1. *For every unlabeled unranked tree $t \in \mathcal{T}(\{a\})$ with $|t| \geq 2$ and integer $k \geq 1$, we have $H_k(t) \leq 2(H^{\deg}(t) + H^{\deg,\ell}_k(t)) + 2\log_2(|t|) + 4$, and $H_k(t) \leq 2(H^{\ell}_k(t) + H^{\ell,\deg}_k(t)) + 2\log_2(|t|) + 4$.*

We note that there exist families of unranked trees over a non-unary alphabet, for which the degree entropy is exponentially smaller than the k^{th}-order label-shape tree entropy. This is not very surprising as the label-shape entropy incorporates the node labels, while the degree entropy does not.

4.3 Labeled unranked trees

In this section, we consider general unranked labeled trees $t \in \mathcal{T}(\Sigma)$ over arbitrary alphabets Σ. The entropies to be compared are $H_k(t)$, $H^{\deg}(t) + H_k^{\deg,\ell}(t)$, $H_k^{\ell}(t) + H_k^{\ell,\deg}(t)$ and $H^{\deg}(t) + H_k^{\ell}(t)$. Somewhat surprisingly it turns out that $H_k^{\ell}(t) + H_k^{\ell,\deg}(t)$ is at most $H^{\deg}(t) + H_k^{\deg,\ell}(t)$ for every tree t:

Theorem 5. *Let $t \in \mathcal{T}(\Sigma)$. Then $H_k^{\ell}(t) + H_k^{\ell,\deg}(t) \leq H^{\deg}(t) + H_k^{\deg,\ell}(t)$.*

Proof. We have

$$
\begin{aligned}
& H_k^{\ell}(t) + H_k^{\ell,\deg}(t) \\
&= \sum_{z \in \Sigma^k} \sum_{a \in \Sigma} n_{z,a}^t \log_2 \left(\frac{n_z^t}{n_{z,a}^t} \right) + \sum_{z \in \Sigma^k} \sum_{a \in \Sigma} \sum_{i=0}^{|t|} n_{z,i,a}^t \log_2 \left(\frac{n_{z,a}^t}{n_{z,i,a}^t} \right) \\
&= \sum_{z \in \Sigma^k} \sum_{a \in \Sigma} \sum_{i=0}^{|t|} n_{z,i,a}^t \log_2 \left(\frac{n_z^t}{n_{z,a}^t} \right) + \sum_{z \in \Sigma^k} \sum_{a \in \Sigma} \sum_{i=0}^{|t|} n_{z,i,a}^t \log_2 \left(\frac{n_{z,a}^t}{n_{z,i,a}^t} \right) \\
&= \sum_{z \in \Sigma^k} \sum_{a \in \Sigma} \sum_{i=0}^{|t|} n_{z,i,a}^t \log_2 \left(\frac{n_z^t}{n_{z,i,a}^t} \right) \\
&= \sum_{z \in \Sigma^k} \sum_{a \in \Sigma} \sum_{i=0}^{|t|} n_{z,i,a}^t \log_2 \left(\frac{n_z^t}{n_{z,i}^t} \right) + \sum_{z \in \Sigma^k} \sum_{a \in \Sigma} \sum_{i=0}^{|t|} n_{z,i,a}^t \log_2 \left(\frac{n_{z,i}^t}{n_{z,i,a}^t} \right) \\
&= \sum_{z \in \Sigma^k} \sum_{i=0}^{|t|} n_{z,i}^t \log_2 \left(\frac{n_z^t}{n_{z,i}^t} \right) + \sum_{z \in \Sigma^k} \sum_{a \in \Sigma} \sum_{i=0}^{|t|} n_{z,i,a}^t \log_2 \left(\frac{n_{z,i}^t}{n_{z,i,a}^t} \right) \\
&\leq H^{\deg}(t) + H_k^{\deg,\ell}(t),
\end{aligned}
$$

where the final inequality follows from the log-sum inequality (Lemma 1). □

As a corollary of Lemma 2 and Theorem 5 it turns out that $H^{\deg}(t) + H_k^{\deg,\ell}(t)$ and $H_k^{\ell}(t) + H^{\deg}(t)$ are equivalent up to constant factors.

Corollary 2. *Let $t \in \mathcal{T}(\Sigma)$. Then*

$$H^{\deg}(t) + H_k^{\deg,\ell}(t) \leq H^{\deg}(t) + H_k^{\ell}(t) \leq 2H^{\deg}(t) + H_k^{\deg,\ell}(t).$$

In the rest of the section we present three examples showing that in all cases that are not covered by Theorem 5 we can achieve a non-constant (in most cases even exponential) separation between the corresponding entropies.

Lemma 7. *(i)* $|t_n| = 2n + 1$,
(ii) $H_k(t_n) \leq \log_2(e) + \log_2\left(n - \left\lfloor \frac{k-1}{2} \right\rfloor\right) + 2$,
(iii) $H_k^{\deg,\ell}(t_n) = 2n$ *and hence* $H^{\deg}(t_n) + H_k^{\deg,\ell}(t_n) \geq 2n$, *and*
(iv) $H_k^{\ell}(t_n) \geq 2n$ *and hence* $H_k^{\ell}(t_n) + H_k^{\ell,\deg}(t_n) \geq 2n$.

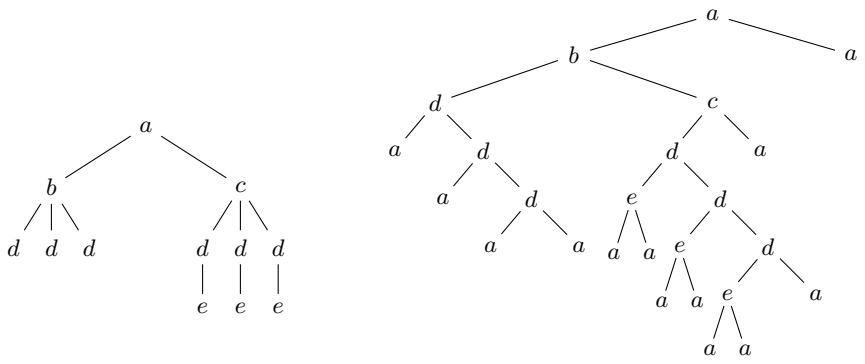

Fig. 2. The binary tree t_3 from Lemma 8 (left) and its first-child next-sibling encoding $\mathrm{fcns}(t_3)$ (right).

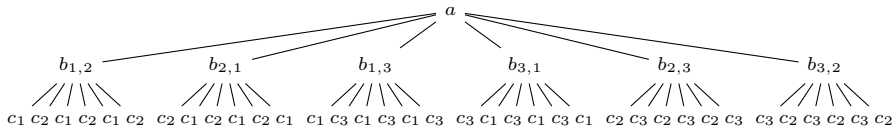

Fig. 3. The tree $t_{3,2}$ from Lemma 9.

For the tree t_n in Lemma 7 one can take $t_n = a(bcbc\cdots bc)$ with n occurrences of b (respetively, c). Lemma 7 shows that there are not only families of binary trees, but also families of unranked (non-binary) trees $(t_n)_{n\geq 1}$ (for which we have to compute $H_k(t_n)$ via the fcns-endcoding) such that $|t_n| = \Theta(n)$ and $H_k(t_n)$ is exponentially smaller than $H^{\deg}(t_n) + H_k^{\deg,\ell}(t_n)$ and $H_k^{\ell}(t_n) + H_k^{\ell,\deg}(t_n)$.

Lemma 8. *There exists a family of unranked trees $(t_n)_{n\geq 1}$ such that for all $n \geq 1$ and $1 \leq k \leq n$:*

(i) $|t_n| = 3n + 3$,
(ii) $H_k(t_n) \geq 2(n - k + 1)$,
(iii) $H^{\deg}(t_n) + H_k^{\deg,\ell}(t_n) \geq 2n$ *and*
(iv) $H_1^{\ell}(t_n) + H_1^{\ell,\deg}(t_n) = 3\log_2(3)$.

For the tree t_n in Lemma 7 one can take $t_n = a(b(dd\cdots d)\, c(d(e)d(e)\cdots d(e)))$ with $2n$ occurrences of d. The tree t_3 is shown in Figure 2. Note that we clearly need $\Omega(\log n)$ bits to represent this tree (since we have to represent its size). This does not contradict Theorem 2 and the $\mathcal{O}(1)$-bound for $H_1^{\ell}(t_n) + H_1^{\ell,\deg}(t_n)$ in Lemma 8, since we have the additional additive term of order $o(|t|)$ in Theorem 2.

In the following lemma, $n^{\underline{k}} = n(n-1)\cdots(n-k+1)$ is the falling factorial.

Lemma 9. *There exists a family of unranked trees $(t_{n,k})_{n\geq 1}$, where $k(n) \leq n$ may depend on n, such that for all $n \geq 1$:*

(i) $|t_{n,k}| = 1 + n^{\underline{k}} + k \cdot n \cdot n^{\underline{k}}$,
(ii) $H^{\deg}(t_{n,k}) + H_1^{\ell}(t_{n,k}) \leq \mathcal{O}(n \cdot n^{\underline{k}} \cdot k \cdot \log k)$ *and*
(iii) $H_{k-1}(t_{n,k}) \geq \Omega(n \cdot n^{\underline{k}} \cdot k \cdot \log(n-k+1))$.

The label set of the tree $t_{n,k}$ is $\{a\} \cup \{b_u \mid u \in [n]^{\underline{k}}\} \cup \{c_i \mid 1 \leq i \leq n\}$, where $[n]^{\underline{k}} = \{(i_1, i_2 \ldots, i_k) \mid 1 \leq i_1, \ldots, i_k \leq n, i_j \neq i_l \text{ for } j \neq l\}$. For $u = (i_1, i_2, \ldots, i_k) \in [n]^{\underline{k}}$ define the tree $t_u = b_u((c_{i_1} c_{i_2} \cdots c_{i_k})^n)$; then $t_{n,k}$ is $a(t_{u_1} t_{u_2} \cdots t_{u_m})$, where $u_1, u_2, \ldots, u_m$ is an arbitrary enumeration of the set $[n]^{\underline{k}}$ (hence, $m = n^{\underline{k}}$). The tree $t_{3,2}$ is shown in Figure 3.

If $k \in (\log n)^{\mathcal{O}(1)}$ then the trees $t_{n,k}$ from Lemma 9 satisfy

$$\frac{H^{\deg}(t_{n,k}) + H_1^{\ell}(t_{n,k})}{H_{k-1}(t_{n,k})} \leq \mathcal{O}\left(\frac{\log k}{\log(n-k+1)}\right) = o(1).$$

This yields a relatively weak separation between $H^{\deg}(t) + H_1^{\ell}(t)$ and $H_k(t)$. In contrast, in Lemmas 7 and 8 we achieved an exponential separation. It remains open, whether such an exponential separation is also possible for $H_1^{\ell} + H^{\deg}$ and H_k. In other words, does there exist a family of trees t_n such that $H_k(t_n) \in \Omega(n)$ and $H^{\deg}(t_n) + H_1^{\ell}(t_n) \in \mathcal{O}(\log n)$?

5 Experiments

We finally complement our theoretical results with experimental data. We computed the entropies $H^{\deg}$, H_k, H_k^{ℓ}, $H_k^{\ell,\deg}$ and $H_k^{\deg,\ell}$ (for $k \in \{0, 1, 2, 4\}$) for 13 XML files from XMLCompBench (`http://xmlcompbench.sourceforge.net`). Table 1 shows the values for H_k, $H^{\deg} + H_k^{\ell}$, $H_k^{\ell} + H_k^{\ell,\deg}$ and $H^{\deg} + H_k^{\deg,\ell}$ (which can be achieved up to lower order terms by compressors). It turns out that for all XML trees used in this comparison the k^{th}-order label-shape entropy (for $k > 0$) from [14] is significantly smaller than the entropies from [12]. In the full version [16, Table 2] the reader finds also the values for H_k^{ℓ}, $H_k^{\ell,\deg}$ and $H_k^{\deg,\ell}$ (divided by the tree size so that the table fits on the page). Additionally, we computed in [16] the label-shape entropy H_k for a modified version of each XML tree where all labels are replaced by a single dummy symbol, i.e., we considered the underlying, unlabeled tree as well (in [16, Table 2] this value is denoted by H'_k). Note again that the label-shape entropy H_k is the only measure for which this modification is interesting. In the setting of unlabeled trees, our experimental data indicate that neither the label-shape entropy nor the degree entropy (which is the upper bound on the number of bits needed by the data structure in [17] ignoring lower order terms; see also Theorem 1) is favorable.

XML	k	H_k	$H^{\deg} + H_k^{\ell}$	$H_k^{\ell} + H_k^{\ell,\deg}$	$H^{\deg} + H_k^{\deg,\ell}$
BaseBall	0	202 568.08	153 814.94	146 066.64	146 066.64
	1	6 348.08	145 705.73	137 957.42	145 323.26
	2	2 671.95	145 705.73	137 957.42	145 323.26
	4	1 435.11	145 705.73	137 957.42	145 323.26
DBLP	0	18 727 523.44	14 576 781.00	12 967 501.16	12 967 501.16
	1	2 607 784.68	12 137 042.56	10 527 690.38	12 076 935.39
	2	2 076 410.50	12 136 974.71	10 527 595.96	12 076 845.69
	4	1 951 141.63	12 136 966.29	10 527 586.31	12 076 836.82
EXI-Array	0	1 098 274.54	962 858.05	649 410.59	649 410.59
	1	4 286.39	387 329.51	73 882.05	387 304.76
	2	4 270.18	387 329.51	73 882.05	387 304.76
	4	4 263.82	387 329.51	73 882.05	387 304.76
EXI-factbook	0	530 170.92	481 410.05	423 012.12	423 012.12
	1	11 772.65	239 499.01	181 101.08	204 649.84
	2	5 049.98	239 499.01	181 101.08	204 649.84
	4	4 345.42	239 499.01	181 101.08	204 649.84
EnWikiNew	0	2 118 359.59	1 877 639.22	1 384 034.65	1 384 034.65
	1	243 835.84	1 326 743.94	833 139.36	1 095 837.20
	2	78 689.86	1 326 743.94	833 139.36	1 095 837.20
	4	78 687.51	1 326 743.94	833 139.36	1 095 837.20
EnWikiQuote	0	1 372 201.38	1 229 530.04	894 768.55	894 768.55
	1	156 710.30	871 127.39	536 365.91	717 721.09
	2	51 557.50	871 127.39	536 365.91	717 721.09
	4	51 557.31	871 127.39	536 365.91	717 721.09
EnWikiVersity	0	2 568 158.43	2 264 856.93	1 644 997.36	1 644 997.36
	1	278 832.56	1 594 969.93	975 110.35	1 311 929.24
	2	74 456.55	1 594 969.93	975 110.35	1 311 929.24
	4	74 456.41	1 594 969.93	975 110.35	1 311 929.24
Nasa	0	3 022 100.11	2 872 172.41	2 214 641.55	2 214 641.55
	1	292 671.36	1 368 899.76	701 433.91	1 226 592.72
	2	168 551.10	1 363 699.16	696 194.53	1 221 474.16
	4	147 041.08	1 363 699.16	696 194.53	1 221 474.16
Shakespeare	0	655 517.90	521 889.47	395 890.85	395 890.85
	1	138 283.88	370 231.89	244 047.64	347 212.36
	2	125 837.77	370 061.20	243 843.87	347 041.31
	4	123 460.80	370 057.77	243 838.09	347 037.86
SwissProt	0	18 845 126.39	16 063 648.44	13 755 427.39	13 755 427.39
	1	3 051 570.48	11 065 924.67	8 757 703.61	10 238 734.83
	2	2 634 911.88	11 065 924.67	8 757 703.61	10 238 734.83
	4	2 314 609.48	11 065 924.67	8 757 703.61	10 238 734.83
Treebank	0	16 127 202.92	15 669 672.80	12 938 625.09	12 938 625.09
	1	7 504 481.18	12 301 414.61	9 482 695.67	9 925 567.44
	2	5 607 499.40	11 909 330.06	9 051 186.33	9 559 968.40
	4	4 675 093.61	11 626 935.89	8 736 301.14	9 285 544.85
USHouse	0	36 266.08	34 369.06	28 381.43	28 381.43
	1	10 490.44	24 249.78	17 968.41	19 438.19
	2	9 079.97	24 037.34	17 569.59	19 216.99
	4	6 308.98	23 634.87	16 830.00	18 783.36
XMark1	0	1 250 525.41	1 186 214.34	988 678.93	988 678.93
	1	167 586.81	592 634.17	394 639.43	523 996.29
	2	131 057.35	592 625.76	394 565.79	523 969.97
	4	127 157.34	592 037.39	393 770.73	523 432.87

Table 1. Values of the four entropies compared in this paper for various XML trees.

References

1. Philip Bille, Pawel Gawrychowski, Inge Li Gørtz, Gad M. Landau, and Oren Weimann. Top tree compression of tries. In *Proc. ISAAC 2019*, volume 149 of *LIPIcs*, pages 4:1–4:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
2. Philip Bille, Inge Li Gørtz, Gad M. Landau, and Oren Weimann. Tree compression with top trees. *Information and Computation*, 243:166–177, 2015.
3. Mireille Bousquet-Mélou, Markus Lohrey, Sebastian Maneth, and Eric Noeth. XML compression via DAGs. *Theory of Computing Systems*, 57(4):1322–1371, 2015.
4. Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (2. ed.)*. Wiley, 2006.
5. Paolo Ferragina, Fabrizio Luccio, Giovanni Manzini, and S. Muthukrishnan. Structuring labeled trees for optimal succinctness, and beyond. In *Proc. FOCS 2005*, pages 184–196. IEEE Computer Society, 2005.
6. Paolo Ferragina, Fabrizio Luccio, Giovanni Manzini, and S. Muthukrishnan. Compressing and indexing labeled trees, with applications. *Journal of the ACM*, 57(1):4:1–4:33, 2009.
7. Philippe Flajolet, Paolo Sipala, and Jean-Marc Steyaert. Analytic variations on the common subexpression problem. In *Proc. ICALP 1990*, volume 443 of *Lecture Notes in Computer Science*, pages 220–234. Springer, 1990.
8. Travis Gagie. Large alphabets and incompressibility. *Information Processing Letters*, 99(6):246–251, 2006.
9. Moses Ganardi, Danny Hucke, Markus Lohrey, and Louisa Seelbach Benkner. Universal tree source coding using grammar-based compression. *IEEE Transactions on Information Theory*, 65(10):6399–6413, 2019.
10. Moses Ganardi, Artur Jez, and Markus Lohrey. Balancing straight-line programs. In *Proc. FOCS 2019*, pages 1169–1183. IEEE Computer Society, 2019.
11. Michal Ganczorz. Entropy bounds for grammar compression. *CoRR*, abs/1804.08547, 2018.
12. Michal Ganczorz. Using statistical encoding to achieve tree succinctness never seen before. In *Proc. STACS 2020*, volume 154 of *LIPIcs*, pages 22:1–22:29. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
13. Adrià Gascón, Markus Lohrey, Sebastian Maneth, Carl Philipp Reh, and Kurt Sieber. Grammar-based compression of unranked trees. *Theory of Computing Systems*, 64(1):141–176, 2020.
14. Danny Hucke, Markus Lohrey, and Louisa Seelbach Benkner. Entropy bounds for grammar-based tree compressors. In *Proc. ISIT 2019*, pages 1687–1691. IEEE, 2019.
15. Danny Hucke, Markus Lohrey, and Louisa Seelbach Benkner. Entropy bounds for grammar-based tree compressors. *CoRR*, abs/1901.03155, 2019.
16. Danny Hucke, Markus Lohrey, and Louisa Seelbach Benkner. A comparison of empirical tree entropies. *CoRR*, abs/2006.01695, 2020.
17. Jesper Jansson, Kunihiko Sadakane, and Wing-Kin Sung. Ultra-succinct representation of ordered trees with applications. *Journal of Computer and System Sciences*, 78(2):619–631, 2012.
18. Markus Lohrey, Sebastian Maneth, and Roy Mennicke. XML tree structure compression using RePair. *Information Systems*, 38(8):1150–1167, 2013.
19. Giovanni Manzini. An analysis of the Burrows-Wheeler transform. *Journal of the ACM*, 48(3):407–430, 2001.

20. Carlos Ochoa and Gonzalo Navarro. RePair and all irreducible grammars are upper bounded by high-order empirical entropy. *IEEE Transactions on Information Theory*, 65(5):3160–3164, 2019.
21. Nicola Prezza. On locating paths in compressed cardinal trees. *CoRR*, abs/2004.01120, 2020.