RJ 10248 (A0207-039) July 26, 2002
Computer Science

IBM Research Report

A Metric for Compression with the Burrows-Wheeler and Move-to-Front Transforms

J. Q. Trelewicz, Cornel Constantinescu, Ron Arps
IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

A Metric for Compression with the Burrows-Wheeler and Move-to-Front Transforms

J. Q. Trelewicz, Cornel Constantinescu, and Ron Arps
IBM Almaden Res. Ctr., 650 Harry Road, San José, CA 95120 USA
e-mail: {trelewicz,cornel}@us.ibm.com, arps@almaden.ibm.com

Abstract. Compression of symbolic information constructed according to a grammar, such as textual strings, has long been a topic of interest in computer science. The Burrows Wheeler Transform (BWT), combined with some form of the Move-To-Front Transform (MTFT), has received significant attention in recent years. Much of the BWT/MTFT literature has debated the suitability of one algorithm flavor over another, justifying these results against standardized test suites. However, there has been little activity in evaluating the input string for suitability with respect to a particular algorithm flavor, without significantly degrading the computational performance of the compression system.

We introduce a metric used to evaluate input strings for BWT/MTFT-type compression algorithm suitability. The metric may be used with input strings of arbitrary length, and may be used with arbitrary input language. The convergence of the metric is discussed, showing that the metric may be approximated accurately with relatively low computational complexity, allowing input to be evaluated with lower complexity than that required for computing the output of the compression algorithm. Furthermore, much of the computation required for the highly parallelizable metric also serves to compute the BWT. As a result, the metric may be employed in practical computational systems.

AMS Subj. Classification: 68W25; 68W10; 94A17

Key Words: String compression, Burrows Wheeler Transform, Move-To-Front Transform

1. Introduction

Compression of symbolic information constructed according to a grammar, such as textual strings, has long been a topic of interest in computer science. The Burrows Wheeler Transform (BWT), combined with some form of the Move-To-Front Transform (MTFT), has received significant attention in recent years. Applying the MTFT after the BWT brings textual strings to a "normalized form", utilizing contextual information inherent in language grammars. Alone, these two transforms do not compress the data but utilize contextual redundancy to facilitate entropy coding. The BWT and MTFT are invertible, allowing lossless compression. The ranks, output from the MTFT, are then entropy-encoded to achieve the needed compression.

Much of the BWT/MTFT literature has debated the suitability of one algorithm configuration over another, justifying these results against standardized test suites. However, there has been little activity in evaluating the input string for suitability with respect to a particular algorithm flavor, without significantly degrading the computational performance of the compression system.

Our application must perform compression in real-time, so that the speed performance of the algorithms are critical, and may sometimes be traded against Compression Ratio (CR). Additionally, the language of compression may not be known in advance, so that methods requiring such prior knowledge are not employed. However, we show that the metric that we introduce in this paper may be used to verify *a-priori* language decisions.

In this paper we review the BWT and MTFT, introducing the notation for these transforms and the metric used in this paper. We give an overview of the recent literature concerning these transforms applied to the compression of symbolic grammatical data. We introduce a metric used to evaluate input strings for BWT/MTFT-type compression algorithm suitability. The metric may be used with input strings of arbitrary length, and may be used with arbitrary input language. The convergence of the metric is discussed, showing that the metric may be approximated accurately with relatively low computational complexity, allowing input to be evaluated with lower complexity than that required for computing the output of the compression algorithm. Furthermore, much of the computation required for the highly parallelizable metric also serves to compute the BWT. As a result, the metric may be employed in practical

computational systems. Finally, we present results obtained by testing our method against files written in several languages, with different alphabet orderings.

2. Background on the Transforms

In this section, we give notation for the paper, describing the BWT and the MTFT.

2.1. Notation. Define the following basic notation:

$$m..n \equiv [m,n] \cap \mathbb{Z}, \quad \text{for} \quad m,n \in \mathbb{Z};\ m \leq n;$$
$$\mathbb{N}_0 = \mathbb{N} \cup \{0\}; \qquad \mathbb{Q}^+ = \mathbb{Q} \cap (0,\infty).$$

For a set E, let $|E| = \text{card}(E)$.

If ζ is an ordered tuple or a sequence, ζ_j is the jth member of ζ, starting with index one. Let $\pi_j(\zeta) = \zeta_j$.

In this paper, we make use of the following small abuse of notation: for some $g : C^1 \times C^2$ with $I^j \subset C^j$, let

$$g(I^1, I^2) \quad = \quad \left\{ g(c^1, c^2) : c^j \in I^j \right\}.$$

2.2. Burrows-Wheeler Transform. Let finite, well-ordered, symbol alphabet A be given, with lexicographical ordering $>$ (extending naturally to $<$, $\leq$, and $\geq$). Define $\$ \notin A$ the sentinel symbol such that $\$ > a\ \forall a \in A$. Denote the elements of A by $\{a^j\}_{j=1}^{|A|}$, where $a^j < a^k$ iff $j < k$. Let $A^+ = A \cup \{\$\}$. Define $\iota : A^+ \to \mathbb{N}$ by $\iota(a^j) = j - 1$ and $\iota(\$) = |A|$.

Let $\mathbf{A}_w = A^w$ (Cartesian product) and $\mathbf{A} = \cup_{w \in \mathbb{N}} \mathbf{A}_w$. For $\alpha \in \mathbf{A}$, let $|\alpha|$ be the length of α in symbols. Define

$$\mathbf{B} \quad = \quad \{s\$: s \in \mathbf{A}\},$$
$$\mathbf{B}_w \quad = \quad \left\{ s\$: s \in \bigcup_{n \in 1..m-1} \mathbf{A}_n \right\} \cup \left\{ s : s \in \bigcup_{n \in 1..m} \mathbf{A}_n \right\}.$$

the set of finite ordered strings.

The Burrows-Wheeler transform (BWT) is denoted by $T_B : \mathbf{B} \to \mathbf{B}$.

The BWT is used to group *locally-frequent* symbols in a string. The transform is invertible, thus suitable for lossless compression of text information. The BWT can be seen as sorting the symbols of the textual string according to their symbol contexts, thus clustering in $T_B(S)$ the symbols that appear in the same context. These substrings, or suffixes, may be ordered through construction of a

suffix array [**1**], essentially a sorted suffix tree. Using conventional algorithms, the suffix tree may be constructed in $O(|S| \log |A|)$ time in $O(|S|)$ space. In [**1**], the suffix array is used primarily for efficient substring searches; however, the calculation of our metric may be seen to progress in a manner analogous to the calculation of the suffix array. Recent interest in the BWT is based not only on these characteristics, making it well-suited for use in natural language string compression, but also on its speed of convergence, which can be shown to be faster on average than that of popular string compression algorithms such as Lempel-Ziv [**2**].

The BWT by itself does not provide compression, but when combined with a transform like the MTFT and an entropy code, compression of the text information may be achieved. Efficient implementations of the BWT are described in, for example, [**3**] and in [**4**].

2.3. Move-to-Front Transform. The Move-to-Front Transform (MTFT) is denoted by $T_M : \mathbf{B} \to \mathbf{R}$, where $\mathbf{R}$ is the set of finite sequences in $\mathbb{N}$. These sequences are called *ranks*, and are used to compress $S \in \mathbf{B}$, by providing position and context information for suffixes of S.

Any localized region of $T_B(S)$ is likely to contain a large number of a few distinct symbols. The overall effect is that the probability that given symbol will occur at a given index j in $T_B(S)$ higher if that symbol occurs near index j in $T_B(S)$. This property is exactly that needed for effective compression by MTFT, which encodes an instance of a symbol by the count of distinct symbols encountered since the most recent occurrence of that symbol. When T_M is applied to $T_B(S)$, the output is ideally dominated by low numbers, which can be efficiently encoded with a Huffman or arithmetic coder.

The MTFT operates in order on the elements of $\kappa = T_B(S)$, giving $t = T_M \circ T_B(S)$. In what follows, $P^j = \{p^{j,k}\}_{k=1}^{|A|}$ is a permutation of A, with P^1 defined as A and

$$\begin{aligned} t_j &= k : p^{j,k} = \kappa_j \\ P^{j+1} &= (\kappa_j, \kappa_1, \ldots, \kappa_{j-1}, \kappa_{j+1}, \ldots, \kappa_{|A|}) \end{aligned}$$

The MTFT can be seen as associating smaller integers with frequently-occurring symbols in S. As mentioned above, T_M does not compress S, but it can help subsequent entropy coding to reduce redundancy. When T_M follows T_B, locally-frequent symbols for every context of

the given string are grouped. It should be noted that T_M is language independent. However, a language-sensitive initial ordering of A can help to improve the CR by reducing the magnitude of t_j by placing frequently-occurring symbols near the beginning of A. In [**5**], a non-lexicographical ordering of A is recommended, which affects the MTFT and thus impacts CR. If the language of the text is known in advance, A may be ordered according to statistics for the language, and this fixed A may be used for all text of that language. The authors of [**5**] display an improvement in CR of less than 1% averaged over their sequence of text files. However, we show in Section 4 that a language-dependent initial ordering of A, when combined with our metric, can be used to verify a language decision, in the sense that the statistics used to order A are tested. Furthermore, although T_M works with mixed languages, mixed language may increase the magnitude of t_j where the statistics of the symbols change, potentially reducing CR.

For fixed $S \in \mathbf{B}$,

$$\begin{aligned} \mathbf{G}_w &= \{(a,\alpha) : a \in A, \alpha \in \mathbf{B}_w, a\alpha \subset S\}, \\ \mathbf{G} &= \{(a,\alpha'\$) : a \in A, \alpha' \in \mathbf{A}, a\alpha'\$ \subset S\}. \end{aligned}$$

For $(a,\alpha),(a',\alpha') \in \mathbf{G}_w$, $\mathbf{G}_w$ is linearly ordered by $>$ applied to α and α'. Note that $\mathbf{G}$ is well-ordered by a relation defined similarly over $\mathbf{G}$. Then T_M operates on this ordered sequence in $\mathbf{G}$ from minimum to maximum member. Sequence $T_M \circ T_B(S)$ will compress well only if T_M produces a highly predictive sequence from $T_B(S)$.

We denote $\pi_2(\mathbf{G}) = \{\alpha^j\}_{j=1}^{|\pi_2(\mathbf{G})|}$, ordered by $>$ so that $\alpha^j < \alpha^k$ iff $j < k$. Intervals $I_{a,j}$ in $\pi_2(\mathbf{G})$ are defined as follows: let $H_a = \{\alpha \in \pi_2(\mathbf{G}) : (a,\alpha) \in \mathbf{G}\}$ and denote the elements of H_a by $\beta^{a,j}$, where $\beta^{a,j} < \beta^{a,k}$ iff $j < k$. Then if $|H_a| = 0$, $I_{a,1} = \mathbf{B}$; otherwise, $I_{a,j} = (\beta^{a,j-1}, \beta^{a,j}]$ for $j > 1$ (noting that there may be no such j), and $I_{a,1} = \{\alpha \in \mathbf{B} : \alpha \leq \beta^{a,1}\}$. Sec. 4.2 develops an example of these intervals for a specific $\mathbf{G}$. Define $H_{a,w}$ and $I_{a,j,w}$ similarly, using τ_w.

Let $\tau_w : \pi_2(\mathbf{G}) \to \pi_2(\mathbf{G}_w)$ be given by

$$\tau_w(\alpha) = \begin{cases} \alpha, & \alpha \in \mathbf{B}_w \\ \alpha^{j_1}, & \alpha^{j_1} \in A^w : \exists \alpha^{j_2} \in \mathbf{B}, \alpha = \alpha^{j_1}\alpha^{j_2} \end{cases};$$

i.e., the w-truncated suffix of α. Note that α^{j_1} is unique for each $\alpha \in \pi_2(\mathbf{G}) \setminus \pi_2(\mathbf{G}_w)$. We may consider τ_w to be a projection, if we consider the elements of $\pi_2(\mathbf{G})$ to be ordered tuples.

The following lemma illustrates the behavior of τ_w as a "projection":

LEMMA 2.1. *For $E \subset \mathbf{B}$ with $|E| < \infty$, $\tau_w(E) \to E$ in $|\cdot|$.*

Proof: Partition E into $\cup_{k\in\mathbb{N}} E_k$ where $e^k \in E_k$ has $|e^k| = k$. Note that $\tau_w(\cup_{k=1}^{w} E_k) = \cup_{k=1}^{w} E_k$ and $|\tau_w(E_k)| \leq |E_k|$ $\forall k$. It follows that $\tau_w(E) \to |E|$ in this measure. □

A number of improvements on the MTFT for specific applications are discussed in the literature. In [**6**], a dictionary of common words in the target language is used, where these words are replaced with alternate strings in $\mathbf{B}$, which in their application provides improvements in the CR of up to 20% over bzip2 (a BWT-MTFT compressor adapted from [**3**]). In [**7**], Inversion Coding (IC) is used instead of MTFT with the BWT, which is shown to provide better CR than BWT and MTFT for *large* files, but comparable results for small files. Our metric (described in Sec. 3) can potentially be used to select between IC and MTFT.

Some implementations avoid the MTFT altogether. In [**8**], symbols in $\mathbf{B}$ are encoded directly, without ranks. This is shown to provide higher CR on large files, but is significantly slower than BWT using MTFT. Our application must run in real-time, so that run-time speed must be traded against compression ratio. Also, the gains in CR obtained coding directly from $\mathbf{B}$ are not significantly high (on the order of 2%) to justify the additional computational complexity for our application. Our metric in Sec. 3 may be used to select between rank-based and non-rank coding on $\mathbf{B}$ in applications where speed performance is not as critical as CR.

2.4. Huffman entropy coding. It is shown in [**9**] that the output of T_B is approximately memoryless and piecewise stationary, making it appropriate for Huffman entropy coding (e.g., [**10**]). We use a fixed Huffman code, in the interest of minimizing computational requirements for our application, and eliminating the need to transmit tables or header information. Additionally, the fixed Huffman code simplifies the analysis. Thus, (potentially assuming a pre-sorted fixed alphabet A) we assume that smaller ranks out of T_M are preferred, since smaller ranks will occur from more frequent prefixes, and should thus be encoded with shorter bit patterns.

Some configurations of the BWT/MTFT algorithms code runs of zeros separately from the entropy coding used on non-zero ranks. To simplify the analysis, we code zeros with the same Huffman code.

Alternatively, the encoder may employ a set of predetermined Huffman codebooks, sending to the decoder the index of the appropriate codebook. The codebook may be chosen by, for example, a metric such as that described in Sec. 3.

Define $\zeta : \mathbb{N}^0 \to \mathbb{Q}^+$, where $\zeta(0) = 0$ and $\zeta(r)$ is the number of bits in the (non-adaptive) entropy code required for a given rank, minus the number of bits required for rank zero (ζ_0).

3. Metric Algorithm

In this section, we develop an algorithm to screen the "suitability" of $T_B(S)$ for T_M; i.e., to indicate those strings S for which higher CR may be achieved. The metric accomplishes this result by measuring the amount of *churning* of the alphabet that will occur when T_M is applied to $T_B(S)$. The metric may also be used to choose between candidate transforms to follow T_B. Let $S \in \mathbf{B}$ be given.

First we define a function that indicates the number of non-equal permutations P^j of A that occur during the calculation of $T_M \circ T_B(S)$. Let $f : \mathbf{G} \to \mathbb{N}_0$ be given by $f(a, \alpha^{a,j}) = |\mathbf{H}_{a,j}|$, where

$$\mathbf{H}_{a,j} = \{b \in A \setminus \{a\} : \exists \alpha \in I_{a,j}, (b, \alpha) \in \mathbf{G}\}$$

and $f(a, \alpha) = 0$ for $\alpha \notin \pi_2(\mathbf{G})$. Let $f_w : \mathbf{G}_w \to \mathbb{N}_0$ and $\mathbf{H}_{a,j,w}$ be defined similarly, using $I_{a,j,w}$. Either of the functions f or f_w may be visualized as a two-dimensional function, such as that shown in Fig. 2.

The development of the metric assumes that large $|f(a, \alpha)|$ coincides with large $|T_M \circ T_B(S)|$. This assumption corresponds to the T_M described above, in that f and f_w give an indication of the number of different P^j that are obtained during the calculation of the MTFT. Implicitly, f and f_w assume an "optimal" initial ordering of A, by the way in which the intervening symbols are counted. Also, $\alpha \in \pi_2(\mathbf{G})$ for which $\int_A f(a, \alpha) da = 0$ indicate intervals of zero runs when applying the MTFT. Alternatively, one may employ a modified T_M that moves symbols toward but not to the front of the alphabet P^j, may need to track whether different or same symbols recur. This information is still present in f, in the number of times that a symbol recurs within an interval I. More details are given in Sec. 4.1.

The structure of the metric allows work performed in calculating the metric to be applied to the calculation of T_B. Specifically, $\forall \alpha$

such that

$$|\{b \in A : (b, \tau_w(\alpha)) \in \mathbf{G}_w\}| > 1,$$

refinement with larger w is needed to generate $T_B(S)$. Those α such that

$$|\{b \in A : (b, \tau_w(\alpha)) \in \mathbf{G}_w\}| \leq 1$$

require no further calculation, since $T_B(S)$ depends only on the lexicographical ordering of the suffixes.

For S, the importance of rapid convergence of f_w to f include the following:

- Because f_w and f lead to $T_B(S)$, calculations which are performed for measurement purposes of the metric effectively calculate the transform. The result is that calculation of the metric does not add much calculation beyond that already required for T_B.
- For some suffixes α, $\tau_w(\alpha)$ may be sufficient to calculate the corresponding elements of $T_B(S)$. This allows fewer calculations to be performed at these indices to obtain $T_B(S)$.
- Because $f_w \to f$, indices for which $\tau_w(\alpha)$ is not sufficient to calculate corresponding elements $T_B(S)$ may be iterated at $w' > w$ without having to discard previous calculations.

The following lemmas are important to developing the rate of convergence of our metric. Since S is finite, it must necessarily converge, but the rate of this convergence is of critical importance to the utility of the wth estimate.

LEMMA 3.1. *Consider α for which $f_w(a, \tau_w(\alpha)) > 0$ for some w. For $w' > w$, if $\exists \alpha^1 \neq \alpha^2 \in \tau_{w'}^{-1}(\alpha)$ such that $f_{w'}(a, \tau_{w'}(\alpha^1)) > 0$ and $f_{w'}(a, \tau_{w'}(\alpha^2)) > 0$, then $\exists a' \neq a$ such that $f_w(a', \tau_w(\alpha)) > 0$.*

Proof: Without loss of generality, it may be assumed that $\alpha^1 < \alpha^2$ and $\forall \alpha^3 \in (\alpha^1, \alpha^2)$, $f_{w'}(a, \tau_{w'}(\alpha^3)) = 0$. Then consider j such that $I_{a,j,w'} = (\alpha^1, \alpha^2]$. Since $f_{w'}(a, \tau_{w'}(\alpha^2)) > 0$, $\exists a' \in A \setminus \{a\}$ such that $\exists \tau_{w'}(\alpha^{j_2}) \in I_{a,j,w'}$ with $(a', \tau_{w'}(\alpha^{j_2})) \in \mathbf{G}_w$. Therefore, $\tau_w(\alpha^{j_2}) = \tau_w(\alpha^2)$, giving the result. □

A consideration important for the convergence of f_w in any sense is that increases in f_w with w are understood and can be shown to disappear with large w.

LEMMA 3.2. *For $w < y$, f_w increases, in the sense that*

$$\int_{\tau_y \circ \tau_w^{-1}(\alpha)} f_y(a, \alpha') d\alpha' > f_w(a, \alpha) \tag{1}$$

iff $\exists \alpha < \alpha^{j_1} < \alpha^{j_2} < \alpha^{j_3}$ *with* $\tau_w(\alpha) \in I_{a,j,w}$ *(the interval with* α *as its left endpoint),* $\tau_w(\alpha^{j_1}) = \tau_w(\alpha^{j_2}) = \tau_w(\alpha^{j_3})$*, and*

$$(a', \alpha), (a, \alpha^{j_1}), (a', \alpha^{j_2}), (a, \alpha^{j_3}) \in \mathbf{G}.$$

Proof: That the conditions imply (1) is clear. So suppose (1), and let $I_{a,j,w}$ be the interval with α as its left endpoint. Denote by $\{I_{a,\ell,y}\}_{\ell=1}^{J}$ the disjoint intervals such that $\tau_w\left(\cup_{\ell=1}^{J} I_{a,\ell,y}\right) \subset I_{a,j,w}$. Eq. (1) implies that $\exists a' \neq a$ such that either

- $\exists \ell \in 1..J$ such that $\exists \alpha^{j_1} \in I_{a,\ell,y}$ with $(a', \alpha^{j_1}) \in \mathbf{G}_y$, but $\forall \alpha^{j_2} \in I_{a,j,w}$, $(a', \alpha^{j_2}) \notin \mathbf{G}_w$. This case contains a contradiction since $\tau_w(\alpha^{j_1}) \in I_{a,j,w}$.
- $\exists \ell^1, \ell^2 \in 1..J$ with $\ell^1 \neq \ell^2$ such that $\exists \alpha^{j_1} \in I_{a,\ell^1,y}$ and $\alpha^{j_2} \in I_{a,\ell^2,y}$ with $(a', \alpha^{j_1}), (a', \alpha^{j_2}) \in \mathbf{G}_y$. Note that the left endpoints of the $I_{a,\ell,y}$ must necessarily map to α under τ_w. This can be seen to be equivalent to the condition stated in the hypothesis.

□

Using the terminology of Lemma 3.2, the lemma states informally that (1) can only occur when τ_w maps suffixes to the endpoint of an interval so that there is a resulting ambiguity as to whether or not $\alpha^{j_2} \in (\alpha^{j_1}, \alpha^{j_3})$.

LEMMA 3.3. *For* $w < y$,

$$\int_{\tau_y \circ \tau_w^{-1}(\alpha)} f_y(a, \alpha') d\alpha' = 0 \tag{2}$$

iff $f_w(a, \alpha) = 0$.

Proof: Since $f_y \geq 0$, (2) holds iff $f_y(a, \alpha^{j_1}) = 0$ $\forall \alpha^{j_1} \in \tau_y \circ \tau_w^{-1}(\alpha)$. Note that either $\tau_w(\alpha^{j_1}) = \alpha$ or $\tau_w(\alpha^{j_1}) \in I_{a,j,w}$ for some j. Thus, (2) holds iff $f_w(a, \alpha) = 0$. □

Measuring the *distance* between suffixes α may be accomplished by mapping onto the interval, then using conventional Euclidean distance measures. Define mapping $\rho : \bigcup_{w \in \mathbb{N}} \mathbf{B}_w \to [0, 1]$ as follows:

$$\rho(\alpha) \quad = \quad \sum_{j=1}^{|\alpha|} \iota(\alpha_j) |A^+|^{-j}.$$

Then ρ is an injective (but not surjective) mapping, giving us the ability to define ρ^{-1} on $\rho(\bigcup_{w \in \mathbb{N}} \mathbf{B}_w)$.

Finally, we define functions δ_w, δ, φ_w, and φ that indicate the transient performance of T_M, giving indications of a potential

change in statistics of S from those expected. Example causes of this change include the following:

- Change in language mid-string, such as when a quote in one language is inserted in a paragraph in another language. This change can be signaled by a series of spikes in φ as the alphabet migrates from one statistical characteristic to another. However, these spikes will not necessarily be contained in an interval in α, unless the alphabet itself changes (such as in a language change from English to Greek).
- Difference in language from that expected, including non-spoken languages such as part numbers, which will not follow conventional language context specifications. This contrast indicates that the incorrect lexicographical ordering for A was used, and is realized in φ as spikes on a significant number of the suffixes, since many of the suffixes will not be compatible with the ordering of A.

What determines the T_M complexity is the magnitude of φ at this a. The particular a at which φ obtains this magnitude is not as important, since the initial ordering of A may be arbitrary. Let $[0,1)$ be divided into disjoint intervals given by $[r_{j_1}, r_{j_2})$ where $r_{j_1}, r_{j_2} \in \rho \circ \pi_2(\mathbf{G})$ and $(r_{j_1}, r_{j_2}) \cap \rho \circ \pi_2(\mathbf{G}) = \emptyset$. Define $\delta_w : [0,1] \to \{0,1\}$ by starting from $\mathbf{G}_w$ to create disjoint intervals $[r_{j_1,w}, r_{j_2,w})$:

$$\delta_w(r) = \begin{cases} 1, & r \in \rho \circ \pi_2(\mathbf{G}) \\ 1, & r \in [r_{j_1,w}, r_{j_1,w} + |A^+|^{-|S|}), \$ \in \rho^{-1}(r_{j_1,w}) \\ 1, & r \in [r_{j_1,w}, r_{j_1,w} + |A^+|^{-w}), w < |S|, \$ \notin \rho^{-1}(r_{j_1,w}) \\ 0, & \text{else} \end{cases}$$

Define δ similarly, with no w contraint. This definition ensures that finite, non-zero-length S may achieve positive measure, and that the weight afforded to $\tau_w(\alpha)$ in σ_w below is proportional to the measure of $\tau_w^{-1} \circ \tau_w(\alpha)$.

Define $\varphi : [0,1] \to \mathbb{N}^0$ by

$$\varphi(r) = \begin{cases} \max_{a \in A} f(a, \rho^{-1}(\alpha)), & r \in \rho \circ \pi_2(\mathbf{G}) \\ \varphi(r_{j_1}), & r \in [r_{j_1}, r_{j_2}), \text{ as above.} \end{cases}$$

Define $\varphi_w : [0,1] \to \mathbb{N}^0$ similarly.

An estimate of the output of the entropy code may now serve as the metric. Let $\sigma_w : [0,1] \to \mathbb{Q}^+$ be given by

$$\sigma_w(S) = \int_0^1 \zeta \circ \varphi_w(r)\delta_w(r)dr,$$

noting the implicit dependence of $\varphi_w(r)$ on S. The function σ is defined similarly. Then, where N is the number of bits required to represent each element of S, the wth estimate of the CR is

$$N/(\sigma_w(S) + \zeta_0). \tag{3}$$

What is required for deriving the rate of convergence of the metric σ_w is an understanding of the natural language properties. It has been shown (e.g., [**11**]) that a natural language can be modeled by an exponential distribution in language features, such as w. We denote the language parameter q in the distribution; i.e., the probability that two suffixes in the language are identical in the first w symbols and different in the $w-1$st is approximately qe^{-qw} for large w.

THEOREM 3.4. *σ_w converges in L_1 for S a natural language string with language parameter q.*

Proof: In L_1, δ_w converges since it is bounded and non-increasing on a finite interval. Specifically, W can be chosen large enough so that $|A^+|^{-w} - |A^+|^{-w-1} < \epsilon$ for $w \geq W$.

Note that $\forall r_{i,y} \in \rho \circ \pi_2(\mathbf{G}_y)$, $\exists r_{j,w} \in \rho \circ \pi_2(\mathbf{G}_w)$ such that $r_{j,w} = \rho \circ \tau_w \circ \rho^{-1}(r_{i,y})$. Let $m(r_{i,y}) = \rho \circ \tau_w \circ \rho^{-1}(r_{i,y})$ and note that $m(r) \leq r$. Furthermore,

$$\min_{r,r' \in \rho \circ \pi_2(\mathbf{G}_y)} |r - r'| \geq |A^+|^{-y},$$

with a similar result for w. Thus, without ambiguity, we can for each $r_{j,w}$, denote by $\{r_{i,y,j}\}_{i=1}^{J_j}$ where $J_j \geq 1$, with $r_{i,y,j} < r_{i+1,y,j}$, such that $m(r_{i,y,j}) = r_{j,w}$. It follows that

$$[r_{i,y,j}, r_{i,y,j} + |A^+|^{-y}) \subset [r_{j,w}, r_{j,w} + |A^+|^{-w}).$$

Let $\epsilon > 0$ be given and suppose that $y > w \geq W$ for some $W > 0$. Let

$$\begin{aligned} R_{j,w} &= [r_{j,w}, r_{j,w} + |A^+|^{-w}), \text{ on which } \varphi_w \text{ is constant,} \\ R_{i,y,j} &= [r_{i,y,j}, r_{i,y,j} + |A^+|^{-y}), \text{ on which } \varphi_y \text{ is constant,} \\ R_j &= R_{j,w} \setminus \cup_{i=1}^{J_j} R_{i,y,j}. \end{aligned}$$

Then

$$\begin{aligned}
\|\varphi_w\delta_w - \varphi_y\delta_y\|_1 &= \int_0^1 \left| \sum_{r_{j,w}} \varphi_w(r)\chi_{R_{j,w}}(r) - \sum_{r_{i,y}} \varphi_y(r)\chi_{R_{i,y,j}}(r) \right| dr \\
&\leq \sum_j \int_{R_j} \varphi_w(r)dr \\
&\quad + \sum_j \sum_{i=1}^{J_j} \int_{R_{i,y,j}} |\varphi_w(r) - \varphi_y(r)|dr \\
&= \sum_j (|A^+|^{-w} - J_j|A^+|^{-y})\varphi_w(r_{j,w}) \\
&\quad + \sum_j \sum_{i=1}^{J_j} |A^+|^{-y}(\varphi_w(r_{i,y,j}) - \varphi_y(r_{i,y,j})) \\
&\leq |A^+|^{-w+1}|\rho \circ \pi_2(\mathbf{G}_w)|
\end{aligned}$$

since it follows from Lemma 3.2 that on the interval $R_{i,y,j}$, $\varphi_w(r) \geq \varphi_y(r)$, and $\varphi_w \leq |A^+|$.

To bound $|\rho \circ \pi_2(\mathbf{G}_w)|$, consider the exponential distribution of the suffixes. The distribution of unique truncated suffixes is likened to the problem of sampling without replacement, in a system where the probability of choosing a particular suffix depends according to the exponential distribution on the other suffixes previously chosen. Thus for $x > |A^+|$,

$$p\left(|\rho \circ \pi_2(\mathbf{G}_w)| = x\right) \leq q\exp(-qx).$$

Integrating over probability,

$$\begin{aligned}
E[|\rho \circ \pi_2(\mathbf{G}_w)|] &= \int_{|A^+|}^{|A^+|^w} xq\exp(-xq)dx \\
&= \left(|A^+|^w q + q^{-1}\right)\exp(-|A^+|^w q) \\
&\quad - \left(|A^+|q + q^{-1}\right)\exp(-|A^+|q)
\end{aligned}$$

It follows that W can be chosen large enough to give the desired result.

The convergence of σ_w follows since the interval of integration is finite. □

It is seen in Theorem 3.4 that σ_w converges as $|A^+|^{-w}$ for large w. This rapid convergence suggests that σ_w can be used as a reasonable approximation to σ, allowing prescreening of S for expected

CR to be performed effectively. Furthermore, since f gives $T_B S$, areas of f_w with ambiguous suffixes can be directly refined to obtain $T_B S$, without performing the whole calculation for f. Specifically, because $T_B S$ can be found after obtaining the number of intervening symbols in each interval, the specific refined r value on the axis need not be calculated once the ambiguity is removed through refinement.

4. Illustrative Examples

This section contains some illustrative examples for definitions given in the text.

4.1. Suffix entropy in f. Note that f captures not only the number of distinct symbols in S, but also the churning of the symbols, which is directly related to suffix entropy. This can be seen in the ordering of suffixes along the ordinate of f. For example, Figs. 1 and 2 show the resulting f from a regular string S with $A = 3$ and $w = 8$, and from a uniformly-distributed random string S_u.

FIGURE 1. Visualization of f, φ, and δ, where $\rho \circ \pi_2(\mathbf{G})$ is the ordinate, increasing to the right, and $A = \{\text{a,b,c}\}$ is the abscissa, increasing up from a. Points are colored with darker color for smaller $|f|$. φ is plotted over δ, where $\delta = 1$ for light gray, and $\delta = 0$ for dark gray. S_r =abcabcabcabcabc.

FIGURE 2. Visualization of f, φ, and δ, in the same style as Fig. 1. S_u =aabaaacbcbbcbac.

The result is that $\sigma = 0.05$ for the regular string and $\sigma = 0.13$ for the random string. This indicates the differences in potential CR for the two strings. For example, if two-bit encoding is used for A, and if the number of bits used for encoding a rank is one larger than the rank itself,

- For S_r, with $A = \{\text{c,a,b}\}$, (3) gives $\sigma = 0.6$ for CR estimate 1.25. Actual compression gives

$$\begin{aligned} T_B S &= \text{ccccaaaaabbbbbc}, \\ T_M \circ T_B S &= (0,0,0,0,1,0,0,0,0,2,0,0,0,0,2), \end{aligned}$$

  for actual CR 1.5. The discrepancy results from the non-uniqueness of suffixes of S_r for $w = 6$.
- For S_u, with $A = \{\text{b,a,c}\}$, (3) gives $\sigma = 0.87$ for CR estimate 1.07. Actual compression gives

$$\begin{aligned} T_B S &= \text{baaabaccbcbbaac}, \\ T_M \circ T_B S &= (0,1,0,0,1,1,2,0,2,1,1,0,2,0,2), \end{aligned}$$

  for actual CR 1.07. It is no surprise that the exact CR is given by (3), since $w = 8$ returns the exact output of T_B; i.e., the suffixes of S_u are unique at length $w = 8$.

4.2. The Intervals $I_{a,j}$. Let A be the English alphabet in lexicographical order, with $S =$"ananab" (the word "banana" reversed). Then $\mathbf{G}$, ordered as $\pi_2(\mathbf{G})$, is given by

$$\{(\text{n,ab\$}), (\text{n,anab\$}), (\text{a,b\$}), (\text{a,nab\$}), (\text{a,nanab\$}), (\text{b,\$})\}.$$

It follows that

$$\begin{aligned} H_a &= \{\alpha \in \pi_2(\mathbf{G}) : (a, \alpha) \in \mathbf{G}\} \\ H_{\text{a}} &= \{\beta^{\text{a},1} = \text{b\$}, \beta^{\text{a},2} = \text{nab\$}, \beta^{\text{a},3} = \text{nanab\$}\}, \\ H_{\text{b}} &= \{\beta^{\text{b},1} = \text{\$}\}, \\ H_{\text{n}} &= \{\beta^{\text{n},1} = \text{ab\$}, \beta^{\text{n},2} = \text{anab\$}\}, \end{aligned}$$

and $H_a = \emptyset \; \forall a \notin \{\text{a, b, n}\}$. Thus, $I_{a,1} = \mathbf{B} \; \forall a \notin \{\text{a, b, n}\}$, and

$$\begin{aligned} I_{\text{a},1} &= \{\alpha \in \mathbf{B} : \alpha \leq \text{b\$}\} \\ I_{\text{a},2} &= (\text{b\$}, \text{nab\$}] \\ I_{\text{a},3} &= (\text{nab\$}, \text{nanab\$}] \\ I_{\text{b},1} &= \{\alpha \in \mathbf{B} : \alpha \leq \text{\$}\} = \mathbf{B} \\ I_{\text{n},1} &= \{\alpha \in \mathbf{B} : \alpha \leq \text{ab\$}\} \\ I_{\text{n},2} &= (\text{ab\$}, \text{anab\$}]. \end{aligned}$$

These intervals may be difficult to visualize. However, if we map each interval by ρ, with $|A| = 26$, $\iota(\text{a}) = 0$, $\iota(\text{b}) = 1$, $\iota(\text{n}) = 13$, we

get the following to four significant places:

$$\begin{aligned}
\rho(I_{a,1}) &= [0, 0.9630]\ \forall a \notin \{\mathrm{a}, \mathrm{b}, \mathrm{n}\} \\
\rho(I_{\mathrm{a},1}) &= [0, 0.07270) \\
\rho(I_{\mathrm{a},2}) &= (0.07270, 0.4816] \\
\rho(I_{\mathrm{a},3}) &= (0.4816, 0.4821] \\
\rho(I_{\mathrm{b},1}) &= [0, 0.9630] \\
\rho(I_{\mathrm{n},1}) &= [0, 0.002692) \\
\rho(I_{\mathrm{n},2}) &= (0.002692, 0.01784].
\end{aligned}$$

Fig. 3 shows f for this simple example, with A restricted to {a,b,n} to simplify the plot. The diagram is warped along the ordinate per mapping by ρ. It can be seen that suffixes are all clustered by starting symbol, resulting from the regular structure of the word "banana". There are two symbols (a,n) leading suffixes before the b-lead string (shown at 0.9630), and there is one symbol (n) leading a suffix before the a-lead strings (shown at 0.07270).

FIGURE 3. Visualization of f, φ, and δ, in the same style as Fig. 1.

Thus, $\sigma = 0.5$ for this string S with $w = 8 \geq |S|$. If eight-bit encoding (e.g., ASCII) is used for the characters of the original string, if the number of bits used for encoding a rank is one larger than the rank itself, the estimate of the CR from (3) is 5.33. Actual compression of S with this ζ and $A = \{\mathrm{n,a,b,}\ldots\}$ in ASCII order gives "nnaaab" from T_B, rank sequence $(0, 0, 1, 0, 0, 2)$ from T_M, for a resulting CR of 5.33. The exact result is obtained since $w \geq |S|$.

4.3. Language Change in S. To demonstrate how σ_w catches changes in language and alphabet clearly, we calculated σ_4 (i.e., $w = 4$) for the passage of text shown in Tab. 1.

Figure 4 shows φ for this passage. The language and alphabet change shows very clearly to the right of the dotted line, where the average value of φ jumps sharply. This increase, which already assumes the optimum pre-ordering of the ASCII character set for this passage, results from the change in symbols mid-passage. What is significant about the detection using σ_w is that the interval of

We sat on hard chairs in the back of the musty bookstore, the smell of dust and old books in the air, the shadows of evening falling through the windows. *She read the words of Marina Tsvetaeva, with a voice clear and bright. And as she read the last stanza,* *“Разбросанным в пыли по магазинам,* *(Где их никто не брал и не берет!),* *Моим стихам, как драгоценным винам,* *Настанет свой черед.”* *I turned to him, thinking to say, how much we are like the dusty books, stacked around us on the shelves. But his eyes were elsewhere, lost in thought as the twilight deepened and the dust settled.*

TABLE 1. Sample multi-language text for test with σ_w.

increased average value in σ_w is related directly to the number of suffixes containing the change in language; e.g., the inclusion of a single word would cause a much smaller impact. This allows language change to be separated from the use of isolated symbols; e.g., Greek letters in mathematical formulae.

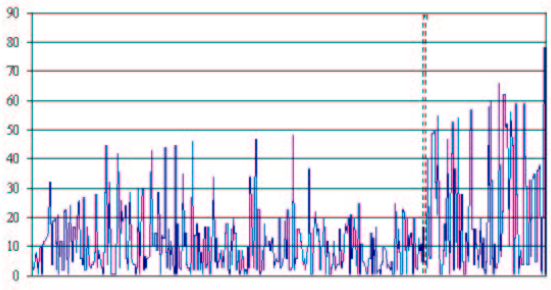

FIGURE 4. Visualization of φ_4, with ordinate $\rho \circ \pi_2(\mathbf{G}_4)$. The dotted line shows compression of the axis for display. The language change is indicated clearly by an increase in the average value of φ_4.

5. Conclusions and Future Study

We have developed a metric for evaluation of natural language strings S for compression with the BWT and MTFT. The calculations required for the metric have been shown to benefit also the

calculation of the transform. The metric allows iterative refinement, with rapid convergence for natural languages.

We are currently evaluating the structure of the space of S, using σ as a metric for the space. This analysis gives insight about the structure of the BWT and about strings in natural languages. This structure can allow comparison with the asymptotic convergence bounds derived in [**12**], independent of model parameters.

References

[1] U Manber and G Myers, "Suffix arrays: a new method of on-line string searches," *SIAM J. Comput.*, vol. 22, no. 5, pp. 935–948, Oct. 1993.

[2] M Effros, "Universal lossless source coding with the Burrows Wheeler transform," in *Proc. DCC'99 Data Compr. Conf.*, Snowbird, UT, USA, 29-31 March 1999, pp. 178–187.

[3] P M Fenwick, "The Burrows-Wheeler transform for block sorting text compression: principles and improvements," *Comput. J.*, vol. 39, no. 9, pp. 731–740, 1996.

[4] B Balkenhol and S Kurtz, "Universal data compression based on the Burrows Wheeler transformation: theory and practice," *IEEE Trans. Comput.*, vol. 49, no. 10, pp. 1043–1053, Oct. 2000.

[5] B Chapin and S R Tate, "Higher compression from the Burrows Wheeler transform by modified sorting," in *Proc. DCC'98 Data Compr. Conf.*, Snowbird, UT, USA, 30 March - 1 April 1998, p. 532.

[6] H Kruse and A Mukherjee, "Improving text compression ratios with the Burrows Wheeler transform," in *Proc. DCC'99 Data Compr. Conf.*, Snowbird, UT, USA, 29-31 March 1999, p. 536.

[7] Z Arnavut, "Move to front and inversion coding," in *Proc. DCC 2000 Data Compr. Conf.*, Snowbird, UT, USA, 28-30 March 2000, pp. 193–202.

[8] A I Wirth and A Moffat, "Can we do without ranks in Burrows Wheeler transform compression?," in *Proc. DCC 2001 Data Compr. Conf.*, Snowbird, UT, USA, 27-29 March 2001, pp. 419–428.

[9] K Visweswariah, S Kulkarni, and S Verdu, "Output distribution of the Burrows Wheeler transform," in *Proc. 2000 IEEE Int'l Sym. on Information Thy.*, Sorrento, Italy, 25-30 June 2000, p. 53.

[10] T M Cover and J A Thomas, *Elements of Information Theory*, John Wiley and Sons, Inc., New York, 1991.

[11] S F Chen, K Seymore, and R Rosenfeld, "Topic adaptation for language modeling using unnormalized exponential models," in *Proc. 1998 IEEE ICASSP, II*, Seattle, WA, USA, 12-15 May 1998, pp. 681–684.

[12] G Manzini, "An analysis of the Burrows-Wheeler transform," *J. ACM*, vol. 48, no. 3, pp. 407–430, May 2001.