Error Resilient LZ'77 Scheme and Its Analysis[1]

Stefano Lonardi
Dept. of Computer Science
University of California
Riverside, CA 92521
e-mail: stelo@cs.ucr.edu

Wojciech Szpankowski
Dept. of Computer Sciences
Purdue University
West Lafayette, IN 47907
email: spa@cs.purdue.edu

Mark Daniel Ward
Dept. of Mathematics
Purdue University
West Lafayette, IN 47907
e-mail: mward@math.purdue.edu

The devastating effect of errors in adaptive data compression is a long-standing open problem. In fact, the non-resilience of adaptive data compression has been a practical drawback of its use in many applications. We shall argue here that practically there is no need for additional overhead in order to correct errors in LZ'77. This seemingly impossible goal is achieved in practice due to the fact that the LZ'77 encoder is unable to decorrelate completely the input sequence.

Our error-resilient LZ'77 encoder is based on the following observation. We observe theoretically and experimentally (cf. [1]) that in a significant proportion of LZ'77 phrases, there is more than one copy of the longest prefix in the compressed file (cf. Theorem 1). More precisely, we define a position i in the text corresponding to the beginning of a phrase to have *multiplicity* r if there exist exactly r matches for the longest prefix that starts at position i. We call M_n the random variable associated with the multiplicity of a sequence of length n generated by a binary memoryless source (with p being the probability of generating "0", and $q = 1 - p$). In Theorem 2 we prove that the random variable M_n follows the logarithmic series distribution (plus some fluctuations), that is, $P(M_n = j) \approx (1/h)(p^j q + q^j p)/j$, where h is the entropy rate.

The positions with multiplicity $r > 1$ are the ones that can be used to embed some of the bits of another binary string, called the *message*. Specifically, the next $\lfloor \log_2(r) \rfloor$ bits of the message will drive the selection of one particular pointer out of the r choices. These additional bits can be used for various purposes such as authentication/integrity or error correction, as described next.

Once the redundant bits of LZ'77 have been identified, we exploit them for channel coding. For error detection and correction, we choose $RS(255, 255 - 2e)$ Reed-Solomon codes. The $2e$ extra parity bytes constitute the message that will be embedded in the redundant bits of LZ'77. The error-resilient encoder first compresses the input sequence using standard LZ'77. The data is broken into blocks of size $255 - 2e$. Then, blocks are processed in reverse order, beginning with the very last. When processing block i, the encoder computes first the Reed-Solomon parity bits for the block $i + 1$ and then it embeds the extra bits in the pointers of block i. We are currently working on a scheme in which e is changed adaptively with the availability of redundant bits in the stream.

We now proceed to sketch the analysis. Let $T_{[1,n]}$ be the first n symbols generated by the source and let L_n be the random variable associated with the length of the longest prefix (phrase) of $T_{[n+1,\infty]}$ which has an occurrence in $T_{[1,n]}$. In other words, the random variable L_n describes the length of the phrases of LZ'77. Henceforth, we use L instead of L_n for simplicity. We associate the variable W_n to the multiplicity at position n in the text, that is, $W_n = \sum_{i=1}^{n-L} \mathbf{1}(T_{[i,i+l-1]} = T_{[n+1,n+L]})$. Using the results by Wyner [3] it is easy to prove the following theorem.

Theorem 1 *Let $T_{[1,n]}$ be generated by a Markov source. Then*

$$\mathbf{E}[W_n] = O(1), \tag{1}$$

that is, the average multiplicity is constant when n is large.

One can define W_n in terms of the associated suffix trie S_n built from the first n suffixes of the text T. In fact, when inserting the $(n + 1)$-st suffix of T into S_n the size of the subtree starting at the branching point of a new insertion is exactly W_n.

In order to study the variable W_n we reduce the problem to a simpler one that, asymptotically, is equivalent to our problem. Instead of analyzing a random suffix tree we construct a random trie built from n *independently generated* strings generated by a memoryless source. It is known that such a trie approximates well the initial suffix trie. (Indeed, if one builds a suffix tree from $n/\log n$ suffixes separated by $\log n$ symbols, then such a tree is asymptotically equivalent to an independently built trie.) As a consequence of this, we can now concentrate our analysis to tries. In such a trie, define M_n to be the size of the subtree starting at the branching point of a new insertion. Then, as we shall prove, M_n and W_n have the same asymptotic distribution. In [2] we establish the following result concerning M_n.

Theorem 2 *Let $z_k = \frac{2kr\pi i}{\ln p}$ for all $k \in \mathbb{Z}$, where $\frac{\ln p}{\ln q} = \frac{r}{s}$ for some relatively prime $r, s \in \mathbb{Z}$. Then*

$$\mathbf{E}[(M_n)^{\underline{j}}] = \Gamma(j)\frac{q(p/q)^j + p(q/p)^j}{h} + \delta_j(\log_{1/p} n) + O\Big(\frac{1}{n}\Big)$$

and

$$\mathbf{E}[u^{M_n}] = -\frac{q\ln(1-pu) + p\ln(1-qu)}{h} + \delta(\log_{1/p} n, u) + O\Big(\frac{1}{n}\Big)$$

where δ_j and δ are periodic functions that have small magnitude, and Γ is the Euler gamma function. We note that δ_j and δ exhibit fluctuation if and only if $\ln p/\ln q$ is rational.

References

[1] Lonardi, S., Szpankowski, W., Joint source-channel LZ'77 coding, *IEEE Data Compression Conference*, 273–282, 2003.

[2] Ward, M. D., Szpankowski, W., Analysis of a randomized selection algorithm motivated by the LZ'77 scheme, *1st Workshop on Analytic Algorithms and Combinatorics*, 2004.

[3] Wyner, A. J., The redundancy and distribution of the phrase lengths of the fixed-database Lempel-Ziv algorithm, *IEEE Trans. Information Theory*, 43, 1439–1465, 1997.

[1] This work was supported in part by NSF Grants CCR-0208709, DBI-0321756 and NIH grant R01 GM068959-01.