Code Drafting, Part 2: Balanced Code Tables

The first article in this series noted that different letters appear with different frequency in text [1]. The frequencies vary with the language and with the type of text.

Analysis of a large body of English text from a variety of sources shows these frequencies, arranged alphabetically:

A	0.08167
B	0.01492
C	0.02782
D	0.04253
E	0.12702
F	0.02228
G	0.02015
H	0.06094
I	0.06966
J	0.00153
K	0.00772
L	0.04025
M	0.02406
N	0.06749
O	0.07507
P	0.01929
Q	0.00095
R	0.05987
S	0.06327
T	0.09056
U	0.02758
V	0.00978
W	0.02360
X	0.00150
Y	0.01974
Z	0.00074

Arranged by decreasing frequency, the list is

E	0.12702
T	0.09056
A	0.08167
O	0.07507
I	0.06966
N	0.06749
S	0.06327
H	0.06094
R	0.05987
D	0.04253
L	0.04025
C	0.02782
U	0.02758
M	0.02406
W	0.02360
F	0.02228
G	0.02015
Y	0.01974
P	0.01929
B	0.01492
V	0.00978
K	0.00772
J	0.00153
X	0.00150
Q	0.00095
Z	0.00074

The importance of letter frequency lies in balancing shaft utilization. The three standard tables given in the first article in this series are significantly unbalanced with respect to the frequencies associated with the shaft pairs.

Here are the shaft-pair frequencies for the three standard tables:

	letters	*shaft pair*	*frequency*
Table 1			
	ABCDEFG	1,2	0.33639
	HIJKLMN	2,3	0.27165
	OPQRSTU	3,4	0.33659
	VWXYZ	4,1	0.05536
Table 2			
	ABCDEF	1,2	0.31624
	GHIJKL	2,3	0.20025
	MNOPQR	3,4	0.24673
	STUVWXYZ	4,1	0.23677

Table 3

AEIMQUY	1,2	0.35068
BFJNRVZ	2,3	0.17661
CGKOSW	3,4	0.21763
DHLPTX	4,1	0.25507

Table 1 is so badly unbalanced that for many strings shaft pair (4,1) would not be used. This does not mean a shaft might not be utilized, since shaft 4 also is in shaft pair (3,4) and shaft 1 also is in shaft pair (1,2). However, for the string

SLIME MOLD

shaft 4 is not utilized.

Tables 2 and 3 also are significantly unbalanced, although less so than Table 1.

It is not difficult to design a frequency-balanced code table. Here are three that are progressively more balanced:

Table 4

EIRUGVQ	1,2	0.31501
TNDMYKZ	2,3	0.25284
ASLWPJ	3,4	0.22961
OHCFBX	4,1	0.20253

Table 5

EIR	1,2	0.25655
TNDUF	2,3	0.25044
ASLMGPV	3,4	0.25847
OHCWYBKJXQZ	4,1	0.23453

Table 6

ETCJXQ	1,2	0.24938
AOIFZ	2,3	0.24942
NSHDB	3,4	0.24915
RLUMWGYPVK	4,5	0.25204

Table 4 was constructed by assigning letters in order of decreasing frequency to shaft pairs in order, cyclically. Thus, E, the most frequently occurring letter, was assigned to shaft pair (1,2); T, the second most frequently occurring letter, to shaft (2,3); and so on. In this method, shaft pair (1,2) has a frequency that is somewhat too high, while shaft pair (4,1) has a frequency that is somewhat too low. Table 4, nonetheless, is more balanced that any of the standard tables.

Table 5 was constructed in a similar fashion, except that a letter was not added if the frequency to that point was greater than 0.25.

Table 6 is the result of a refinement to the procedure for constructing Table 5. In building Table 6, a letter was not added to a shaft pair if it would make the frequency to that point greater than 0.25. If this could not be done for any shaft pair, the letter was arbitrarily added to shaft pair (4,1).

Note that no matter how balanced a code table is, it can be defeated by cleverly chosen strings. For example,

JAZZ ALIVE

does not use shaft pair (3,4) of Table 6. However, because shafts 3 and 4 are in other shaft pairs, all shafts happen to be utilized in this example.

Reference

1. *Code Drafting, Part 1: Introduction*, Ralph E. Griswold, 2004:
http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_cd1.pdf

Ralph E. Griswold

Department of Computer Science
The University of Arizona
Tucson, Arizona