5 Recognizing Plaintext: The Log-Weight Method for Bigrams

In the last four sections we used only the single letter frequencies of a natural language. In other words, we treated texts as sequences of independent letters. But a characteristic aspect of every natural language is how letters are combined as bigrams (letter pairs). We may hope to get good criteria for recognizing a language by evaluating the bigrams in a text. Of course this applies to contiguous text only, in particular it is useless for the polyalphabetic example of Sections 3 and 4.

In analogy with the LW score we define a **Bigram Log-Weight (BLW) score** for a string. Let p_{ij} be the probability (or average relative frequency) of the bigram $s_i s_j$ in the base language. Because these numbers are small we multiply them by 10000.

Tables containing these bigram frequencies for English, German, and French are in `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/8_Transpos/Bigrams.html`

In contrast to the single letter case we cannot avoid the case $p_{ij} = 0$: some letter pairs never occur as bigrams in a meaningful text. Therefore we count the frequencies k_{ij} of the bigrams $s_i s_j$ in a string $a \in \Sigma^r$, and define the BLW-score by the formula

$$S_2(a) := \sum_{i,j=1}^{n} k_{ij} \cdot w_{ij} \quad \text{where } w_{ij} = \begin{cases} \log(10000 \cdot p_{ij}) & \text{if } 10000 \cdot p_{ij} > 1, \\ 0 & \text{otherwise.} \end{cases}$$

Note. We implicitly set $\log 0 = 0$. This convention is not as strange as it may look at first sight: For $p_{ij} = 0$ we'll certainly have $k_{ij} = 0$, and setting $0 \cdot \log 0 = 0$ is widespread practice.

To calculate the BLW score we go through the bigrams $a_t a_{t+1}$ for $t = 1, \ldots, r-1$ and add the log weight $w_{ij} = \log(10000 \cdot p_{ij})$ of each bigram. This approach is somewhat naive because it implicitly considers the bigrams—even the overlapping ones!—as independent. This criticism doesn't mean that we are doing something mathematically wrong, but only that the usefulness of the score might be smaller than expected.

We prepare matrices for English, German, and French that contain the relative frequencies of the bigrams in the respective language. These are in the files `eng_rel.csv`, `ger_rel.csv`, `fra_rel.csv` in the directory `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Files/` as comma-separated tables. The corresponding bigram log-weights are in the files `eng_blw.csv`, `ger_blw.csv`, `fra_blw.csv`. Programs that compute BLW scores for English, German, or French are `BLWscE.pl`, `BLWscD.pl`, and `BLWscF.pl` in the Perl directory.

As an example we compute the scores for the Caesar example, see Table 16. The correct solution is evident in all three languages.

Table 16: *BLW scores for the exhaustion of a* Caesar *cipher*

```
BLW scores English    German      French
  FDHVDU    1.4        3.1         2.2
  GEIWEV    5.8 <---   7.3 <===    4.3
  HFJXFW    0.9        0.3         0.0
  IGKYGX    2.2        2.1         1.3
  JHLZHY    0.5        1.9         0.3
  KIMAIZ    5.9 <---   5.2         4.9
  LJNBJA    1.1        2.4         0.9
  MKOCKB    2.7        4.2         0.8
  NLPDLC    3.0        2.8         1.4
  OMQEMD    3.5        3.8         3.6
  PNRFNE    3.6        4.7         3.6
  QOSGOF    5.8 <---   4.0         3.4
  RPTHPG    4.5        2.6         2.7
  SQUIQH    2.3        0.6         6.3 <---
  TRVJRI    4.1        4.3         4.9
  USWKSJ    3.3        3.7         2.0
  VTXLTK    1.3        2.0         1.1
  WUYMUL    3.1        2.9         2.7
  XVZNVM    0.6        1.3         1.0
  YWAOWN    5.5        2.3         0.0
  ZXBPXO    0.0        0.0         0.0
  AYCQYP    3.2        0.0         0.3
  BZDRZQ    1.0        2.1         1.1
  CAESAR    7.7 <===   7.5 <===    8.4 <===
  DBFTBS    4.7        3.5         0.6
  ECGUCT    5.5        3.6         5.5
```