A The Ideographic Description algorithm

The Ideographic Description algorithm, defined by the Unicode Consortium, describes a way to represent a grapheme by its components. All Han logographs (*i.e.* graphemes) can be recursively decomposed into smaller components that are themselves logographs. With IDS denoting an logograph, the Ideographic Description algorithm can be written as

```
IDS := IDS
 | BinaryOperator IDS IDS
 | TrinaryOperator IDS IDS IDS/
```

This simply means that an logograph can be decomposed into one, two or three smaller logographs. The operators indicate the relative positions of the operands. Many logographs can be described in more than one way using this algorithm as the logographs can themselves be broken down further.

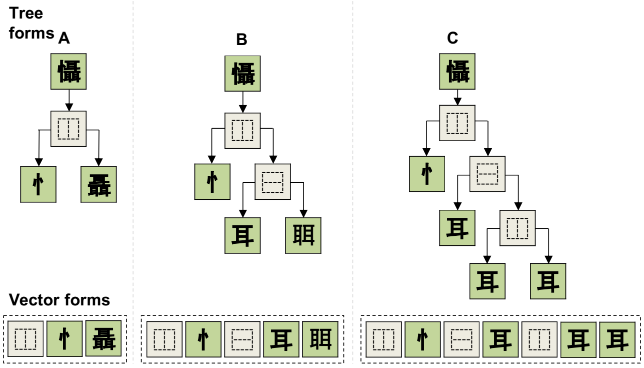

Figure S5: (reproduced from Figure 1) ***A***, ***B*** *and* ***C*** *are equivalent ideographic description sequences for the same logograph. Each sequence can also be represented as a tree.*

Figure S5 shows three different ways the logograph for “admire” can be decomposed at different levels of granularity. The granularity depends on the set of basic logographs at which the algorithm terminates. As the algorithm is recursive, the decomposition of a logograph is a tree or a sequence with the operators evaluated in prefix order. The sequence representation is lossless as it preserves the relative geometric position between the components. The logograph can be reconstructed perfectly from the sequence of components.

Figure S5 also shows how the three sequence of components are represented as three vectors of count. Different from the sequence representation, representing the components as a vector of counts is lossy, as the geometric relationship between components are not preserved. Representing graphemes as sequences rather than as vectors may lead to higher prediction accuracy if the positional information is useful for the task.