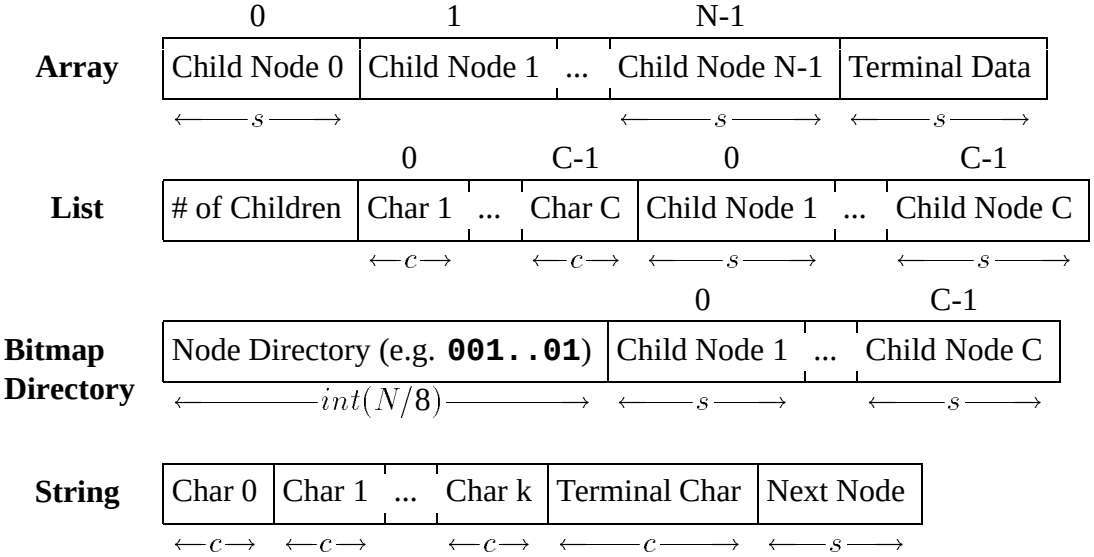
Keyword Dictionary Compression Using Efficient Trie Implementation

Toshiyuki Masui
Center for Machine Translation, Carnegie Mellon University

A *Keyword Dictionary*, a set of pairs of a string and its attribute, is used in many applications like spell checking and information retrieval, where content adressability is required. Although hash tables are widely used, a *trie* structure is more appropriate for those applications if it can be implemented in a compact and efficient way. We propose a method to construct a compact and efficient trie structure which can be applied to both large- and small- scale keyword dictionaries.

We use four types of trie node representations eclectically depending on the number of children of the node, to minimize unused spaces and to provide fast access. Those representations are array representation, bitmap directory representation, list representation, and string representation. When the number of alphabets is 30, each representation is selected when the number of children nodes is 29–30, 4–28, 2–3, and 1, respectively.

In the figures below, N denotes the number of characters (alphabets) used in the dictionary where the ordinal of each character c ($ord(\mathrm{c})$) has a value between 0 and $N-1$, and C denotes the number of children of a node. Each character is stored in c bytes in the memory, and each pointer to the child node is stored in s bytes.



We get better results than the trie creation method proposed by Aoe (J. Aoe., "An efficient digital search algorithm by using a double-array structure," *IEEE Transactions on Software Engineering*, vol. 15, pp. 1066–1077, Sept. 1989.)