INTEGRATING DIVERSE KNOWLEDGE SOURCES IN TEXT RECOGNITION

Sargur N. Srihari, Jonathan J. Hull and Ramesh Choudhari

Department of Computer Science
State University of New York at Buffalo
Amherst, New York 14226

ABSTRACT

The capabilities of present commercial machines for producing correct text by recognizing words in print, handwriting and speech are very limited. For example, most optical character recognition [OCR] machines are limited to a few fonts of machine print, or text that is handprinted under certain constraints; any deviation from these constraints will produce highly garbled text. This paper describes an algorithm for text recognition/correction that effectively merges a bottom-up refinement process that is based on the utilization of transitional probabilities and letter confusion probabilities, known as the Viterbi algorithm [VA], together with a top-down process based on searching a trie structure representation of a lexicon. The algorithm is applicable to text containing an arbitrary number of character substitution errors such as that produced by OCR machines.

The VA is a method of finding the word that maximizes likelihood over all possible letter combinations and not necessarily those in a lexicon; it is based on a dynamic programming formulation which leads to a recursive algorithm. The algorithm can be viewed as a shortest path algorithm through a directed graph called a trellis. The negative of the log transitional probabilities are associated with the edges of the trellis and the negative log confusion probabilities are associated with the nodes. The cost of a path is then the sum of all the edge and node values in the path. Some generalizations of the VA have used either a fixed but small number of alternatives per letter of the input word, called the Modified Viterbi Algorithm [MVA] or a variable number of alternatives per letter; these alternatives can be determined by the letters that have the highest confusion probability.

The VA (and its variations) is a purely bottom-up approach whose performance may be unacceptable due to the fact that the resulting strings do not necessarily belong to a dictionary. One approach to this problem is to use top-down contextual information, in the form of a dictionary of allowable input words, to aid the bottom-up performance of the VA.

The trie data structure is suitable for determining whether a given string is an initial substring, or prefix, of a word in the lexicon. Essentially, the trie considers words as ordered lists of characters, elements of which are represented as nodes in a binary tree. Since the VA proceeds by computing for a given length the most likely prefix, the trie is an efficient data structure.

Simultaneous search of the VA trellis using a variable number of alternatives per input letter and the trie structure is achieved using a binary array A. Element A[j,i] is set to 1 if the jth letter of the alphabet is a possible correction for the ith letter of the input word, and 0 otherwise. Thus the paths of the trellis that need to be evaluated are only those that begin at the 1's of the first column of A and proceed through the 1's of the subsequent columns. Before evaluating a path that proceeds from one column of A to the next column, that path is determined to be legal with respect to the trie. The computational complexity of the resulting algorithm is of the same order as the VA.

Experimental results with the algorithm are described for a data-base of 6372 words of garbled English text using a lexicon of 1724 words represented as a trie of 6197 nodes. With first-order letter transitional probabilities and the optimum number of alternatives per letter (with respect to execution time) the

algorithm was able to correct 87% of the garbled words. This is in contrast to correction rate of 35% with a purely bottom-up approach and 82% with a purely top-down approach. As an example, the garbled sentence

"If we looi at what has produsedlomputer imtelligence go far, we see multiple lamers, each of which rests on primitives of the naxd tayfr dowm, forminc a hierarcfical structure with a great deal interposed between the intelligent prphvem and the transistors which ultimatelu suppodt it"

is corrected by the algorithm to

"If we look at what has produced -------- intelligence so far, we see multiple layers, each of which rests on primitives of the next sites down, forming a hierarchical structure with a great deal interposed between the intelligent program and the transistors which ultimately support is",where the hyphens indicate a rejected word. Rejections can be reduced (or eliminated) by increasing the number of alternatives per letter.

Thus the algorithm is able to effectively utilize top-down knowledge in the form of a lexicon of legal words, channel characteristics in the form of a table of letter confusion probabilities and two-types of bottom-up information: letter shapes represented as vectors and letter transitional probabilities.