Project Title: Automatic Abbreviation Generation

Name: Gregg Stum, MS; Patrick Demasco MS; Kathleen McCoy, PhD

Address: Applied Science and Engineering Laboratories
University of Delaware/ A.I duPont Institute
PO Box 269
Wilmington, DE 19899

Telephone: (302) 651-6830

Sponsor: National Institute on Disability and Rehabilitation Research,
U.S. Department of Education,
400 Maryland Ave. SW,
Washington, DC 20202,
Director: William Graves, Ed.D.

Nemours Foundation
A.I. duPont Institute
PO Box 269
Wilmington, DE 19899
Research Director: Charles Hartzell, Ph.D.

References: Stum, G. M. (1991). Automatic Abbreviation Generation. Technical Report 92-01. Dept. of Computer and Information Sciences, University of Delaware.

Stum, G. M., Demasco, P. W., and McCoy, K. F. (1991). Automatic Abbreviation Generation. Proceedings of the RESNA 14th Annual Conference (Kansas City, MO). Washington, D.C.: RESNA PRESS. 97 - 99.

Key Words: Abbreviation Expansion, Augmentative and Alternative Communication

Category: Independent Living Aids - Communication Methods and Systems

Purpose

The goal of this project is to increase the communication rate of physically disabled individuals via abbreviation expansion. Currently, abbreviation expansion systems require the assignment of a unique abbreviation to each word that its user abbreviates. The user must then memorize all of these assignments. Flexible Abbreviation Expansion, one application of Automatic Abbreviation Generation, seeks to eliminate these requirements. In this approach the expansion system recognizes any well-formed abbreviation for a word, eliminating the need to define and memorize specific abbreviation assignments.

Progress

Automatic Abbreviation Generation is demonstrated with a prototype Flexible Abbreviation Expansion system implemented in C++. The prototype distinguishes among 9 different rules that it uses to categorize the possible expansions of an abbreviation. It has been tested on vocabularies of up to 1300 words. It is source-code portable between both Unix and DOS, and compiles under a variety of distinct compilers. The prototype generates all of the abbreviation information it requires and stores this in memory. Under Unix, it consumes as much memory as it needs. Under DOS, it is sensitive to the amount of available memory and restricts its generation accordingly.

Methodology

The expansion system reads the vocabulary that the user wishes to abbreviate, generates a table comprising all of the abbreviations that it expects to see from the user for each word in the vocabulary, and identifies the abbreviation rules that each of the abbreviations represent with respect to their word. The user is then free to enter any well-formed abbreviation for the intended word. The system looks up this abbreviation in the table of anticipated abbreviations and categorizes the possible expansions according to the rule that the abbreviation represents. For example, the abbreviation ‘ab’ is a *truncation* for the words ‘absent’, ‘abundant’, and ‘abbreviation’, and is a *strict-contraction* for the word ‘aplomb’. The list of words is then ordered according to preferences based on features of the word (e.g. its associated rule). The most likely expansion is taken as the word with the highest preference.

While of significant importance, the user interface is considered separate from the expansion system. That is, the expansion system provides all of the possible expansions for an abbreviation, and the user interface determines how to report these to the user. This allows a wide variety of designs that can be tailored to any particular user. One possibility is for the interface to simply replace the abbreviation with the most preferred expansion. The user would then press a special <reject> key to get the next most preferred expansion. Another possible design is for the interface to present a list of the most preferred expansions, from which the user directly selects the one intended. A third possibility is a mix of the two, where the expander replaces the abbreviation with the most preferred expansion, and presents a list of expansions if the user rejects it.

Results

The prototype demonstrates that an abbreviation expansion system can recognize well-formed abbreviations of words. This project is being continued by GMS Systems with a Small Business Innovation Research Grant from the Department of Education. The objective of this grant is to demonstrate the feasibility of using Flexible Abbreviation Expansion to enhance job opportunities. The grant supports the development of a more practical implementation that rates the possible expansions by the user’s preferences, bases its information in a file rather than memory, and incorporates a user-interface that can support a wide variety of configurations such as those described above. Ultimately the information used by the Expander is seen to be included in a CD-ROM language module having an on-line dictionary or encyclopedia.