PROOFREADER An Interim English Text Proofreader

A collection of subsystems and data bases for doing proofreading of English text on Altos is currently under construction. These subsystems will hardly venture into the semantic gulf at all. They will simply read a piece of text and decide whether each string of letters in that text is an English word. For the near future they will make no attempt to correct a non-word into a word, but will simply bring all the non-words to the user's attention.

At the moment, the collection consists of three programs and one massive data base. One program, BTREETEST, is a data base interrogation and maintenance program which operates about at level 1 (the author can run it, and you might be able to learn to run it too). Another program, PROOFREADER, actually uses the data base to proofread a piece of text. Since it only requires one parameter and operates non-interactively, we could probably call it a level 2 program (you could run it with ease the first time) even now. The third program, SORTDICT, sorts MW-format files (of which appendix B is one) alphabetically. It is of no use to BTREETEST or PROOFREADER, but may be of use to you. The data base is a giant Alto file (called PAGING.PG) containing an encoded English dictionary with about 35000 English words. It corresponds roughly to the New Merriam-Webster Pocket Dictionary, except that the definitions have been left out, and a number of technical terms and names and the like have been added. The programs and data base together consume something like 1500 pages of Alto disk space.

The programs will now be presented in greater detail, and the format of a dictionary entry set forth so that if you have a few favorite words of your very own, you can add them to your copy of the dictionary. L'Academie Parc reserves the right to control the content of the main dictionary, but what you add to (or delete from) your own copy is your business.

1. PROOFREADER

The program is invoked by

PROOFREADER TEXTFILE OUTPUTFILE DICTFILE

where OUTPUTFILE currently defaults to "QSPELL.TX" and DICTFILE currently defaults to 'PAGING.PG". It reads through the text file and associates the characters together into atoms (roughly corresponding to normal word boundaries) which are looked up in the dictionary file. Some words are not present in the dictionary because they are inflected forms of words which are present: e.g. "dogs". If the program fails to find an exact match in the dictionary, it then investigates various ways in which the atom might have been formed by inflecting a word which is in the dictionary. Atoms known to be correctly-spelled English words, as well as atoms known not to be English words, are stored in a hashed LRU cache to speed the look-up process.

If words include embedded punctuation or spaces ("e.g.", "a la carte") then the original decisions about where to break the text into atoms may have been in error. If the surrounding context is contained in the dictionary, then the word is considered correctly spelled. For example, the dictionary contains the following entries:

"e: #.g."
".g: e#."
"a: # la carte"
"la: a # carte"
"carte: a la #"

At the end of processing (or perhaps earlier in a very long text file with many questionably-spelled words) the program produces a list of questionably-spelled words on OUTPUTFILE. The list is ordered according to first occurrence, and following each word is the number of times (since the word was last mentioned, if ever) that the word appeared in the text. For purposes of the output listing, two words are considered the same only if their capitalizations and spellings are identical.

It seems to be difficult to take advantage of the supposed fact that in English, only proper nouns and the first words of sentences or sentence fragments are capitalized. In fact, at least in the Parc documents proofread thus far, the general practice seems to be to capitalize anything that needs emphasis. Issues of capitalization are currently being punted by providing two switches:

/A says that any atom which begins with a capital letter is by definition correctly spelled. This avoids having personal names and places flagged as questionably spelled. It also means that the first word of each sentence is not checked.

/C says that unless an atom begins a sentence (immediately follows a ".", "?", or "!") if it begins with a capital letter it must have the attribute "proper noun" in the dictionary.

The default case is that words which begin with small letters must be something other than a proper noun, but everything else is acceptable.

For your added proofreading pleasure, a modification of USER.CM is available which includes a P quit switch for Bravo, which causes PROOFREADER to be invoked on the file whose name is in Buffer 3, after which Bravo is re-invoked with two windows, one for the questionable spellings and one for the original file.

2. BTREETEST

The dictionary is stored on the disk in the form of a B-Tree with variable-length records (for background on B-Trees, see Knuth, vol. 3, pp. 473-479). Each dictionary entry is in the form of a BCPL-format string followed by one 16-bit word which encodes the inflection classes to which the word belongs. The program BTREETEST is a rather fragiledictionary maintenance program build on top of some very nice B-Tree maintenance subroutines which I shall be glad to release if anyone asks. BTREETEST has an interactive command structure, including the following commands:

N(ame of paging file:)
FILENAME Allows the user to specify a dictionary file other than PAGING.PG. May be done only when dictionary file is closed. Do you really want to do this?

I(nitialize and open tree...)
Creates and opens an empty tree, which can subsequently be added to by various commands. Do you really want to do this?

O(pen pre-initialized tree...)
Opens the dictionary file, reads buffers into memory.

M(erriam-Webster input from file:)
FILENAME Reads MW-format entries from FILENAME, and modifies the dictionary file accordingly. More about MW format in section III.

F(ind keys:)

KEY Displays the greatest key in the dictionary less than or equal to the one you typed. LF or CR advances to the next key ad nauseam. N or DEL terminates the command. The "Info" field may be decoded using Appendix A if you persevere.

S(how page: #)

OCTALNUMBER This enables one to trace out the actual structure of the B-Tree. Starting with the page number printed out by the Open, one can wend his way down through the tree. Notice, by the way, that the keys in the root of the tree seem to be considerably shorter than those in the leaves. This is no accident. I plan to write a paper about it when I get a chance.

C(lose tree...)

Closes the dictionary file, writes out dirty pages. Extremely important to do if you have modified the dictionary.

Q(uit)

Return to the command processor.

In the all-too-likely event that something goes wrong and you want to quit and start over, it is very important to close the tree (thereby writing out dirty pages) if you have modified it at all. If you are not in a position to do this, you can call BTREETEST from SWAT, then say C Q, and finally control-K to SWAT.

3. MW Format

This is the format of the files from which dictionary entries are built and modified. Each such file consists of a sequence of entries, not necessarily in alphabetical order. Changes are made in sequence order.

An entry consists of the following:

[WORD ATR1 VAL1 ATR2 VAL2 ... ATRn VALn]

where there may be no space between the left bracket and the word, and there must be at least one space between the last value and the right bracket. Blanks within the word are represented by the character "@". The dictionary is used by several groups, and many of the attributes which are useful in other applications are ignored in this one. The following attributes and values are used by this application:

DELETE	*	Delete this dictionary entry
N	S	This is a common noun which pluralizes by adding "s". (tool, toy)
N	ES	This is a common noun which pluralizes by changing a final "y", if any, to "i" and adding "es". (church, flunky)
N	*	This is a common noun which is irregular, or normally spoken of in the plural, or whose plural doesn't make sense. (man, men, analyses)
N	FALSE	Any common noun meanings for this word should be deleted.

V	S-ED	This is a verb which adds "s" and "ed" and "ing". (talk, fool)
V	S-D	This is a verb which adds "s" and "d" and drops an "e" before adding "ing". (use, revise)
V	S-#ED	This is a verb which adds "s" and doubles its final consonant before adding "ed" or "ing". (clap, trot)
V	ES-ED	This is a verb which adds "es" and "ed" and "ing" (box, crouch)
V	*	This is an irregular verb form.
V	FALSE	Any verb meanings for this word should be deleted.
ADJ	R-ST	This is an adjective which gets stronger by adding "r" or "st". (wide)
ADJ	ER-EST	This is an adjective which gets stronger by adding "er" or "est". (tall)
ADJ	*	This is an adjective which doesn't get stronger (joyful).
ADJ	FALSE	Any adjective meanings for this word should be deleted.
ADV	ER-EST	This is an adverb which gets stronger by adding "er" or "est". (fast)
ADV	*	This is an adverb which doesn't get stronger (joyfully).
ADV	FALSE	Any adverb meanings for this word should be deleted.
NPR	S	This is a proper noun which pluralizes by adding "s". (Alto)
NPR	ES	This is a proper noun which pluralizes by adding "es". (Jones)
NPR	*	This is a proper noun which is irregular, or normally spoken of in the plural, or whose plural doesn't make sense.
NPR	FALSE	Any proper noun meanings for this word should be deleted.
NPR	(SIC S)	This is a proper noun which is capitalized exactly as shown and which pluralizes by adding "s". (OISystem)
NPR	(SIC ES)	This is a proper noun which is capitalized exactly as shown and which pluralizes by adding "es".
NPR	(SIC *)	This is a proper noun which is capitalized exactly as shown and which is irregular, or normally spoken of in the plural, or whose plural doesn't make sense.
NPR	(SIC FALSE)	Any proper noun meanings for this word, capitalized exactly as shown, should be deleted.
COMP	*	These are all other words which do not inflect. My data structure lumps them all together as "OTHERPART *"
CONJ	*	
DET	*	
NUMBER	*	
ORD	*	
PREP	*	
PRO	*	

PUNCT *
QUANT *
PREFIX *
INTJ *
SPECIAL *

OTHERPART FALSE Any "other part" meanings for this word should be deleted.

Included for your amusement as Appendix B is a page randomly chosenfrom a file called XEROXWORDS.DICT. It is sorted not for BTREETEST's convenience, but for the reader's.

By the way, it would be appreciated if you would keep separate .DICT files for names, for technical jargon, and for English words which are just missing from the dictionary. This would facilitate my incorporating these words into the main dictionary later. A great many in- and un- and re- and -able and -ly, etc., words are missing from the dictionary. Techniques similar to those of Kaplan and Kay may later be employed to disassemble prefixes and suffixes, in order to reduce the necessary size of the dictionary and accommodate the productivity of English.

4. SORTDICT

This subsystem sorts MW-format files alphabetically (appendix B is a fragment of a MW-format file). This can be useful if you are trying to eliminate duplicate entries or want to print them. BTREETEST could hardly care less whether its MW-format files are sorted. To sort, you say

SORTDICT INPUTFILE OUTPUTFILE

INPUTFILE and OUTPUTFILE can be the same.

5. How to Get All This Wonderful Stuff

The file <ALTO>PROOFREADER.DM contains PROOFREADER and BTREETEST and SORTDICT and their symbol files. The dictionary tree file is in mode binary on <MCCREIGHT>PARCENGLISH.TREE. You should read it into your Alto as PAGING.PG. If you are just interested in trying it out and don't feel like using 1500 pages on your disk to do it, a model 31 disk with all the goodies on it is available, which you are welcome to borrow or copy.

Appendix A

```
structure DE:   // Dictionary Entry

    [ Key:  [ Length byte   // Number of bytes in key
              1
            Char ,1 byte
            ]
      Info word       // The coding
      ]
```

```
//      Coding of parts of speech in the Info field

manifest
        [
        ImproperNoun = #3
        NS = #1
        NEs = #2
        NOther = #3

        Verb = #34
        VSEd = #4
        VEsEd = #10
        VSXed = #14
        VSD = #20
        VOther = #24

        Adj = #140
        AjRSt = #40
        AjErEst = #100
        AjOther = #140

        Adv = #600
        AvErEst = #200
        AvOther = #400

        ProperNoun = #3000
        NPS = #1000
        NPEs = #2000
        NPOther = #3000

        SicNoun = #14000
        NSS = #4000
        NSEs = #10000
        NSOther = #14000

        OtherPart = #20000


        AnyNZValue = #177777
        ]
```

Appendix B

```
[BCPL
        NPR (SIC *) ]
[became
        V * ]
[benchmark
        N S ]
[bias
        N ES
        V ES-ED ]
[biaxial
        ADJ * ]
[bibliog.
        N *
        SUBSTITUTE ((bibliography)) ]
```

```
[binary
        ADJ * ]
[binder
        N S ]
[bipolar
        ADJ * ]
[bode
        V S-D ]
[boded
        DELETE * ]
[bootstrap
        V S-#ED ]
[boule
        N S ]
[Bravo
        NPR * ]
[breadboard
        N S
        V S-ED ]
[broad
        ADJ ER-EST ]
[brush
        N ES
        V ES-ED ]
[buffer
        V S-ED ]
[Burroughs
        NPR * ]
[byte
        N S ]
[ca.
        ADJ *
        SUBSTITUTE ((circa)) ]
[CACM
        NPR (SIC *)
        SUBSTITUTE ((communications of the association for
                computing machinery)) ]
[callee
        N S ]
[capacitance
        N S ]
[capacitor
        N S ]
[CCD
        NPR (SIC S)
        SUBSTITUTE ((charge-coupled device)) ]
[CDC
        NPR (SIC *)
        SUBSTITUTE ((control data corporation)) ]
[cei
        DELETE * ]
[cf.
        V *
        SUBSTITUTE ((compare)) ]
[ch.
        N *
        SUBSTITUTE ((chapter)) ]
[checkout
        N *
```

ADJ *]
[checksum
N S]
[cholesteric
ADJ *]