XIII International Conference on
Salishan Languages
University of Victoria
August 17-19, 1978

A COMPUTERIZED CONCORDANCE/DICTIONARY

Timothy Montler
University of Hawaii

The linguist working with his own language has immediate, random access to an infinite amount of data. He is his own informant, and his native intuition is his file cards, notebooks, text collection, and tape recorder. Those who work on languages that are not their own and especially those who deal with the more exotic languages are at a distinct disadvantage. Not only do they have limited access to informants, but they also have an onus of bookkeeping that grows in proportion to the success of their fieldwork. Any subsequent analyses of the data will require further seeking, sorting, and collating of relevant forms and contexts. In this age of high quality, portable, sound recording equipment it is possible to collect an immense amount of linguistic data in a relatively short time. However, this deluge of information may actually hinder the linguist's productivity in that more time must be spent in organization and bookkeeping leaving less time for the exercise of his special abilities, transcription and linguistic analysis.

In this paper I wish to illustrate an example of how the linguist's data organization burden can be greatly alleviated with the help of a computer. The particular example I will discuss here is Anthony Mattina's computerized concordance and dictionary of the Colville language. My part in this project included writing and debugging the programs that produced the concordance and intermediate stages of the dictionary. In what follows I will outline the development of the concordance and dictionary noting the analytic linguistic uses that have already been made of it and some that are planned.

The input of the data began in September 1977. Five long texts along with a large amount of miscellaneous file card and notebook data collected by Mattina in several years of fieldwork were entered into the University of Hawaii's IBM 370/158 disc storage by means of an IBM 2741 terminal. To the user there is little difference between entering data into the computer this way and simply retyping it with an ordinary IBM typewriter. All Colville data were entered in the conventional Salish orthography using the Camwil 721M typing element. Each line was entered with a serial number and with an interlinear English translation. (See Figure 1)

Figure 1 is an illustration of the input format. Mattina had already done extensive morphemic analysis of the data allowing him to enter the words with morphemes separated by hyphens. Each morpheme to be collated by the concordance program was preceded by a mark, roots by a '/' and particles and affixes by a '*'.

It was necessary to have a word in the English to match each word in the Colville so that the program could properly align the collated word with its translation. When more than one English word corresponded to one Colville word they were connected by hyphens. When the Colville word had no direct English translation a dummy word consisting of hyphen bounded by spaces was entered. So, each pair of lines had the same number of words, or, in the eyes of the computer, the same number of spaces.

After each set of data was entered, a command typed at the terminal would run a program that checked each pair of lines for a mismatch in number of words. It also checked for blank lines and serial numbers out of order. When errors were found messages were printed at the terminal indicating their nature and location so that they could easily be found and corrected.

Looking ahead to what form the final output would be, it was decided that printing out a thousand page concordance on the typewriter terminal would be impractical and expensive, typing up the terminal for days. It was therefore necessary to transliterate the original Colville input into an orthography that could be printed by the computers high-speed line printer. Since we were still free to use upper and lower-case characters, the changes were minor, leaving the new orthography quite readable. The line printer prints an entire line at once. This allows printing time to be cut to a few hours but precludes back-spacing. So, all glottalized consonants were replaced with the consonant followed by an apostrophe; x̣ was replaced by capital X; ƛ̓ by T'; ʷ by capital W; ə by e; ʕ by g; and ? by 7.

After a data set was entered and the mismatch checking program run another command entered at the terminal would execute a program that performed the transliteration. This program was also able to spot certain typographical errors such as the non-Colville phonemes b and š, for example, Such an error would cause an appropriate message indicating line number and type of error to be printed at the terminal. The errors were then corrected and the program run again for a double check.

The original Colville input was not discarded but carried along as a third interlinear line so that each pair was now a triple consisting of the original, the transliterated, and the English version. (See Figure 2)

By December the texts and most of the file card material had been entered. A preliminary concordance was made of this much data. 80,000 examples of 1,800 roots and 350 affixes were collated in this 2,000 page print out. Figure 3 is an illustrative sample of the concordance.

The same data were processed by several slightly modified versions of the concordance program. Each of these 'spur' programs was designed to pick out a particular phenomenon that was both linguistically interesting and explicitly definable in terms of the structure of the input format. For example, one program collated all of the compounds and their glosses. A compound was definable as any word with more than one root mark (/). Other programs found all reduplicated roots, all roots with a vowel in final position, and all words with more than one full vowel. Samples of these are shown in figure 4.

By the first of this year the second stage, that of converting the concordance into dictionary format, had begun. As the concordance was printed, it was also recorded on magnetic computer tape. Small parts of it, 50 pages or so at a time, were then copied into the direct access disc storage. These parts could then be displayed at the terminal and edited 'on-line'. The editing included mainly entering glosses for the roots, sub-entries under each root, and marks to show which parts of the example sentences were to be deleted by a subsequent program. As the special type ball was not needed for this data entry, the editing was done at a CRT (cathode-ray tube) terminal. The CRT terminal is a video screen connected to a typewriter keyboard. Its advantage over the regular terminal is that it is much faster. Lines or parts of lines can be inserted, deleted or changed with simple commands and an entry can be found and displayed on the screen in seconds.

Only one or two example sentences were needed for each entry in the dictionary. The concordance showed several, sometimes hundreds of examples of a root. It was, therefore, necessary to delete most of the example sentences. Also, it was often the case that only parts of a sentence were relevant to the entry.

The decisions as to which parts of which examples would be kept were made on-line at the terminal. A special character (the vertical bar, '|') was inserted at the beginning and end of the section of sentence to be saved. The actual deletions were then done by a program run on the entire dictionary.

The final product is to be a root dictionary with each root as a main entry and the several words in which it commonly appears sub-entries. Many of these sub-entries will themselves have sub-entries. Each entry (main and sub-) has a gloss, and etymological and grammatical information. These were inserted at the terminal in a way that minimized the amount of purely organizational information that needed to be entered. The program that did the deletions also performed the repetitive dictionary structuring tasks such as labeling one line a main entry, another a sub-entry etc.

The dictionary was organized to conform to the conventions for computerized dictionaries developed by Professor Robert Hsu and others at the University of Hawaii Linguistics department and Social Science Research Institute. These were described by Sharon Mayes for the Thompson dictionary in Working Papers for the XI International Conference on Salish Languages (1976)

At this point in time the dictionary is still one or two years from completion. The work has concentrated on the roots so that most of the major entries are finished. However, the affixes and particles have barely been touched. These will eventually be entered as main entries like the roots. Although there are fewer affixes than roots, there were more examples of each collated by the concordance. Another problem in dealing with the affixes is that one affix may have several allomorphs sorted to far separate places in the concordance while a single long entry may contain several homophonous affixes.

One of the great advantages of computerized dictionary making is that sorting is quick and easy. The most often used program was one that resorted the main entries and the sub-entries within those, according to the special Colville alphabetical order. This program will be useful in disambiguating homophonous entries since it sub-sorts on the gloss. If a different gloss is inserted for each homophone in a large entry the reordering program will pull them apart and make separate, contiguous entries of them.

Another important feature available for a computer processed dictionary is a 'finderlist'. One of the commonly used programs developed by Robert Hsu generates an inverted dictionary i.e. an English-Colville dictionary from a Colville-English dictionary. Key words in the glosses have been marked for extraction by the finderlist program. These then become main entries in the inverted dictionary. This will also require some on-line editing.

The final edited version of the dictionary with roots, affixes, and particles merged and with a complete finderlist will be sent on computer tape to a composer to be prepared for publishing.

Intermediate stages of the concordance/dictionary have been saved in printed form and on tape. The complete concordance and spur concordances will be used for further research in Colville linguistics.

Dr. Mattina has already made extensive use of the concordance in preparing a paper on the Colville transitive suffixes. He was able to inspect the occurrence of the suffixes with many roots and other suffixes, and then check the behavior of these roots in other environments. The concordance is expected to be extensively used in the future for work on the Colville intransitives, reduplication, ablaut, etc.

The concordance program was originally written specifically for this project, but it has since been rewritten so that it is more generally usable. Professor Hsu has designed a modular concordance kit called MODUCON. The various functions of the original concordance have been rewritten as modules to be plugged into the MODUCON program.

General concordance programs have been problematic in that everyone who wants to make a concordance has very specific and different ideas about what form the input data is to be in and what the final output is to look like. The Colville concordance output, for example, was to be in a format compatible with the dictionary processing programs, and the input was trilinear. Programs written to meet the needs of every concordance maker tend to be expensive to run and cumbersome to use.

The MODUCON program attempts to escape these problems by allowing the user to write or have written small ad hoc sections of programs. The modules then become part of the larger concordance program. MODUCON is thus flexible enough for practically any type of concordance. It is, however, still somewhat cumbersome for the non-programmer to use. It is expected that this problem will lessen as the library of modules grows.

A final technical note: all programs were written for the SPITBOL compiler of the SNOBOL4 programming language and run on the IBM 370/158 at the University of Hawaii Computing Center.

INPUT

7. cù-s-əlx: "uɬ *s-*c-/ʔkín*-x, *mət stim̓ kʷ *n-/ac̓ənt*-íls

7. They-said-to-him: - "What's-the-matter, maybe something - is-bothering-yo

*uɬiʔ kʷu a-*k-*s-/xʷìst*-aʔx *uɬiʔ kʷu a-*k-*s-/ɬwín*-əm.

that - you-are-going-away-from-us and - you-are-leaving-us.

Figure 1.

7. cu-s-elx: "uL *s-*c-7ki'n-x, *met stim' kW *n-ac'ent*-i'ls

*uLi7 kWu a-*k-*s-/xWist*-a7x *uLi7 kWu a-*k-*s-/Lwi'n*-em.

Figure 2.

```
A FC98.                                                s-n-wl-a7st-i'p. //
C                                                       Boiled-eggs. //
A BJ115.                                        mi sic t-kW'u'l'-a7st-m-elx, t-kW'el-kW'ul'-s i7 cq
C                                              Then  -  they-fix-the-feathers-on, they-fix the arro


.HW  *-a7t
A CP96.    w-s-elx, meL ixi7 cus-elx i7 s-yag-p-cin-s i7 c-cam'-a7t i7 s-qi'lxW, ixi7 T'xW-en-t-is i7
C          he-tells-them what       his-trouble-is-to the  little - people,  and-then he-kills the ma
A FD767.                                               Li'l-a7t. //
C                                                       Rain-sprinkle. //


.HW  *-a7tk
A CP244.   'c-x-s-elx, way' axa7 c-kic-x-s-elx ixi7 i7 t s-enkW'-enkW'-es-pin-a7tk i7 t kWel'-kWl'a'l
C          rs-got-there,   - - they-got-to-him - some  -  yearling - - calves,   - cows. //    - Then


.HW  *-a7x
A bw7.     -7ki'n-x, met stim' kW na-c'ent-i'ls uLi7 kWu a-k-s-xWi'st-a7x uLi7 kWu a-k-s-Lwi'n-em. //
C          be  something -   is-bothering-you  that   -  you-are-going-away-from-us and - you-are-le
A FC875.                                               c-en-pe-pi'lx-a7x. //
C                                                       The#re-fixing-to-go-in-on-the-rolls. //
A CP127.   -k'l'-i'p: "sta7 lu't, ala7 t'i kmix ixi7 ken c-m'ay7-x-t-wi'xW-a7x, uLi7 kWu .n-sgay-cen-
C            Coyote> .  $Heck no,    here  -  only -  -  telling-stories-to-one-another-we-are,  an
A bw272.    "way' L-c-kic-x i-s-qWsi'7, uL way' gapna7 p i-k-c-yag-mi'x-a7x, kWu k-s-7i'ckn-a7x, way'
C          e-said> .  - $He-got-back my-son, and - now -  I-want-you-all-to-gather,  we are-going-to
A GW354.   x-men-t-em, uL iwa7 k'eL-7exW-7exW-kWu-kst-em k-s--en-xWst-i'tkW-a7x. //  way' uL c-k'i't-
C          came-a-little-closer, -  -      a-begging-him  to-walk-in-the-water. //    - -        She-
A FC238.                                               k-s-c'i'nt-a7x. //
C                                                       What-could-he-say. //
A FC236.                                           ken k-s-c'i'nt-a7x. //
```

Figure 3.

CP236.	s-kem'-s-qlaw'-m	'he-kept-his-money'	/kem'	/qlaw'
GY98.	n-pkW-L-T'a'q-na7-m	'he-put-it-in-his-pocket'	/pkW	/T'a'q
GW243.	i-s-wi7-s-ga'c'-em	'I'm-done-reading'	/wi7	/ga'c'
FC897.	nkW'-s-pin-tk	'one-year'	/nkW'	/pin

COMPOUNDS

NB218.	miw's-u'l'axW	'half-way'
NB184.	a-s-yag-p-ci'n	'you-are-hard-up-for'
NB240.	k-La7-p-i'n'k	'get-into-close-shooting'
NB207.	7asl-i'wL	'two-boats'

FORMS WITH MORE THAN ONE FULL VOWEL

FC467.	k'ra-m	'swim'	/k'ra
BW18.	c-kWi-s	'he-took'	/kWi
BJ37.	kWni-m	'he-took'	/kWni
BW171.	lakli'-s	'she-locks-it'	/lakli'

ROOTS WITH V /_#

GW202.	s-pa7-pa7s-i'nk-s	'he-started-to-feel-bad'
GW215.	k-yw-yw-i'na7-lx	'they-have-good-hearing'
GW241.	s-t-t7i'w-t-a7x	'the-youngest-one'
GW345.	a-k-s-kW'ec-kW'act	'your strength'

REDUPLICATED FORMS

Figure 4.