Work Papers of the Summer Institute of Linguistics, University of North Dakota Session

Volume 16 Article 1

1972

A new language data processing system for field linguists

James D. Ashworth
SIL-UND

Recommended Citation
Ashworth, James D. (1972) "A new language data processing system for field linguists," *Work Papers of the Summer Institute of Linguistics, University of North Dakota Session*: Vol. 16 , Article 1.
DOI: 10.31356/silwp.vol16.01
Available at: https://commons.und.edu/sil-work-papers/vol16/iss1/1

A New Language Data Processing System for Field Linguists

James D. Ashworth

The present generation, beginning perhaps 25 years ago, has often been referred to as a generation of revolution. Certainly one of the most subtle yet pervasive of these in its effect on American culture is the rapidly increasing sophistication and economical use of automatic large-scale information processing systems-commonly called the "computer revolution". However, the large capital expenditures required to make such systems operational, the limited memory capacities of economical machines, and the levels of special expertese required of system designers and (in too many cases) of non-technical would-be users (students, clerks, and businessmen), strongly biased the use of computers during the 50's and early 60's toward solution of tedious numerical problems in science and engineering as well as the bookkeeping chores of business. These were activities for which there were suitably explicit procedures, a high promise of pay-off in either fresh knowledge or a savings in labor costs, and adequate financial backing.

During the 50's and early 60's there also were linguists such as Zellig Harris who tried to develop explicit linguistic theories to the degree that even problems as difficult as translation and library cataloging could be "put on the machine" and done automaticly. Although an amazing amount of money was spent, these attempts were largely unsuccessful for natural language. However, the computer's requirements of explicitness did help to motivate, or at least condition, far-reaching theoretical developments in context-free, transformational, and stratificational grammar, which in turn are being used in trying to develop suitably expressive but highly formal languages for man-machine communication. These more mundane tools may well serve man's needs in language information processing better than could any direct attempt at whole-hog "computerization of language".

Paul Garvin (1962; 385) mentions three ways that computer systems might participate in linguistic research: 1) as a kind of "bookkeeper" of linguistic facts such as meanings and coocurrences of morphs, 2) as a "user" of linguistic theory for such ends as mechanical translation and information retrieval, and 3) as a "user" of linguistic methodology to work out theoretical results. I know of no successful projects at all of the third kind, and although there have been many interesting results in the second of Garvin's three areas, I know of only a few carefully constrained information retrieval systems (such as Dow Chemical's chemical library system) which are clearly preferable to non-theoretically based, less costly counterparts.

The computer as a kind of short-sighted but fast, reliable bookkeeper has enjoyed a much greater success than it has in either of the other two of Garvin's roles. Word counts for stylistic and content analysis, machine aids for compiling dictionaries and word lists, automated concordance projects, and non-theoretically based information retrieval systems (such as the Educational Resource Information Center system) have saved researchers large amounts of time and money as well as providing an otherwise virtually unattainable degree of exhaustiveness and reliability.

Most computer systems processing natural language texts have been designed with specific short-term goals of a researcher in mind. A field linguist, however, needs a varity of information presented in a compact orderly way that will continue to be useful as his attention shifts among the various aspects of the language being described. Concordance projects such as the one supervised by Joseph E. Grimes and Archie M. Kahan of SIL and the University of Oklahoma Research Institute respectively provide a valuable service to linguists needing to quickly check all the environments in which various words and morphemes occur. However, Grimes' and Kahan's system does not seem to be intended to display semantic and the subtler grammatical aspects of the texts it processes as well as it shows cooccurrences of words and morphs.

Quoting Garvin (1962; 385,6)

> "The purpose of language data collection is to collect linguistic data in a systematic way in order to make them available for convenient inspection by the researcher. The

most common form of this is the compilation of concordances. Without extensive syntactic processing of the text, concordances are limited to the inclusion, together with the word of interest, of a specified number of additional words to the left or right in the immediate neighborhood, or of all the words reaching in both directions from the given word to a particular punctuation mark. The usefulness of concordances is unquestioned, but from the researcher's standpoint they constitute no more than an organized file of raw data.

"A form of language data collection which does more data processing for the linguistic researcher than a concordance is the automatic compilation of dictionaries from texts with interlinear translation. This is a computer application which to my knowledge has not yet been tried, but which it would be relatively simple to implement from a programming standpoint. It consists in effect of not much else than the alphabetizing of every form of the original text together with the interlinear translation that goes with it. Assuming that the original has been consistently transcribed, there will now be an alphabetic file of the words of the text, and the researcher will be limited to essentially two tasks: 1) to decide how many of the alphabetized words will be part of the same lexical unit and therefore included in the same dictionary entry, and what is to be chosen as the canonical form into dictionary definitions. It is clear that this is merely the beginning of lexicographic work, if more than a simple word list is intended. For more extensive lexicographic efforts, it might be possible to combine the program suggested above with a concordance program where the researcher can use the output to expand this dictionary entries by the information contained in the concordance.

It is important in this connection to consider what are the prerequisites for such a computer application. To be possible at all, language data collection requires a text segmented into words or other units equivalent to words. This means in effect that the automatic processing of (say) medieval manuscripts without indication of word boundaries presupposes at least enough

> pre-editing to insert word boundaries and punctuation. Needless to say, the processing of text recorded from nonliterate languages has to meet the same requirements. In addition, the problem of consistency arises with utmost severity in both cases.

A few years after Garvin's article appeared, Lamb & Gould (1964) published some results of their experimentation with computer-generated concordances. Their book itself has not been available to me to date, but a review by Grimes (1965) describes it in terms that suggest to me that they may have been influenced by Garvin's comments quoted above. Their system reads cards on which strings of words from a text have been keypunched and, with them, other cards on which parallel strings of word-glosses have been keypunched, and it prints out a concordance, each word shown in its contexts, with interlinear word glosses so that a linguistic investigator is enabled, not only to make use of the concordance's cooccurrence information more readily, but also to observe the kind of consistencies found among his meaning hypotheses in varying contexts. Thus Lamb & Gould's system would seem to provide an important aid for bilingual dictionary construction. Unfortunately, it is not clear whether their system is currently available to field workers and, if so, on what terms.

It seems to this writer that there are missing from the systems mentioned above at least four features that would be desirable and feasible in a concordance system for field linguists. First, it would be desirable to provide more contextual data; often the context conditioning the occurrence of a word extends for twenty or more word positions. (Consider the use of the word "they" in the second sentence of the preceding paragraph.) Second, it is apt to be desirable to decrease the volume of printout generated since both computer time and shipping can be costly. In the systems mentioned above the average word-occurrence is reprinted in a separate line somewhere once for each of the word positions in an average line. Thus, if on the average there were twelve words (including context) shown on a line in a concordance, there would be twelve other lines in the concordance also showing it for that one occurrence. That means twelve times as much printing time besides the increased time to sort the data into the proper order.

Third, any system allowing simple changes to be made simply certainly has an advantage over one presupposing pure data to begin with. Not only clerical errors, but errors of linguistic judgement can seriously clutter up dictionaries and concordances long after they are discovered if making corrections seems to challenge the researcher's technical expertese in an area peripheral to his main objectives. On the other hand, when changes to the text data can be made easily and naturally, using a computer concordance system may help sharpen an early analysis without arousing fears that an initial error rate of 10% would require a virtual re-do of the whole input.

Fourth, a system of computer programs that is not closely tied to one specific type of computing machinery is likely to be useful to more people in more situations, even though perhaps less efficient than a system of computer programs tailored to a specific installation of computing hardware. At least for a decentralized, international, low capital-base organization like SIL, the benefits in flexibility would be expected to far outweigh any increased costs in computer time. Of the many "machine independant programming languages" FORTRAN and COBOL are in widest use. Since FORTRAN is designed primarily for numerical problems and COBOL for handling large collections of data in a variety of formats, COBOL seems the logical choice for the "language" in which our liberated computer system is to be formulated. And finally, the linguist may wish to indicate more word structure than just word and morpheme boundaries. He may also wish all forms of specific types to be grouped together in his concordance or dictionary.

The foregoing five benefits, along with certain features of Grimes & Mahan's and Lamb & Gould's concordance systems, have been incorporated by this author into a system of computer programs now well along in development. The result is technically far from revolutionary, but hopefully it may have some part in bringing a private revolution to some now-illiterate tribesmen because of the vision of Richard S. Pittman who suggested several of its salient features, and described its potential usefullness.

I call this system the Text Morphology Display System. It produces an inter-linearly translated text printout and a morpheme concordance-dictionary from morph data cards and text note cards (described below) which have been prepared by the linguist. The concordance-dictionary does not directly show contexts, but rather lists index numbers (the morph-card sequence numbers) which uniquely identify each morph-occurrence in the text. Either the text display or the concordance-dictionary is available separately, but when used together each can make the other more useful. For example, when particular morphs in particular contexts are being studied (from the text display printed by the system), the linguist is apt to be helped by noting the varity of meanings (and cognates) assigned to them in other contexts as shown by the concordance-dictionary. Likewise, when studying apparantly related forms with the use of the concordance-dictionary, information concerning their distribution may be gained by comparing their contexts as shown in the text at each usage location mentioned in the concordance-dictionary. It seems that in many languages linguistic distinctiveness of a morph is shown more strongly in its beginning segments than in its final segments (R. S. Pittman, conversation). Thus the sort programs associated with the Text Morphology System are normally set up to sort in the usual left-to-right "alphabetical" style. If a linguist wishing to use this system finds that distinctiveness in his language normally increases from left to right instead, he may want to ask for the corresponding change in the sorting technique used for making the concordance-dictionary. This is often an easily made change, but depends on the particular installation where the programs are to be run.

Data enters the Text Morphology Display System by means of standard 80-column "IBM" data processing cards keypunched in either the morph-data format or the text-note format. Morph-data cards are the basic units of the system giving location, morph spelling, cognate, and gloss for each individual morph occurrence. Text notes are primarily intended for giving sentence glosses. When the cards are read by the computer, they are arranged into the order in which they are to occur in the text display and then are merged with similar text data stored on magnetic tape from a previous computer run. Each card entering the system needs a sequence number to specify its position in the text display as well as to make future corrections and merging of text data possible. The linguist may keypunch a sequence number into columns 1-8 of a card, or he may leave columns 1-7 blank, in which case the computer will provide a sequence mumber by adding oooooo2 to the value contained in columns 1-7 of the previous card. If column 8 is blank, zero will be substituted; otherwise column 8 will not be changed. Column 8 serves not only as a part of the sequence number; it also signals whether the card is in the morph-data or the text-note format. If column 8 is punched numeric or left blank, it will be considered a morph-data card, otherwise a text-note card.

Morph data cards provide columns 9-24 to spell out the morphs, columns 25-40 for a cognate, and columns 41-80 for glosses and other grammatical data. Two successive blanks within any of these groups of columns terminates the data of that kind to be shown in the text display. All morph-card data, however, will appear in the concordance dictionary. Text-note cards appear only in the text display and there as simple card images. All 80 columns are printed in a line between the constructed lines of preceding and following morph-card data.

The Text Morphology Display System provides the entire text as context to each morph occurrence. The volume of printout is very low for the number of forms cited since in the text display each morph takes up only a fraction of a line and no more than one line in the concordance-dictionary. Frequently occurring forms with identical morph spelling, cognate, and gloss data are cited four per line in the concordance-dictionary.

Changes to wrong data can be made by submitting cards with sequence numbers which match those of the erroneous ones. On such correction cards only data of the type to be changed must be repeated. The other kinds should be left blank. If on the card whose sequence number is 10014800 there is some mistake in the cognate data, which let us say should be SYAL, the correction card will contain:

```
column 1                column 25
↓                       ↓
10014800b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸SYALb̸b̸b̸...
```

(b̸ here marks a blank space on a card) If the information on Morph data card (say) 1001480L needs simply to be removed, that may be done by submitting 1001480L*D*E*L*E*T*E*b̸b̸b̸... To merge in new data, simply submit cards with the new data and assign their sequence numbers uniformly throughout the interval into which the cards are to go. It is important to initially assign sequence numbers far enough apart so that insertions can be made easily. That is why system-assigned sequence numbers are separated by an interval of 20.

The Text Morphology Display System is written almost entirely in the COBOL computer programming language. Special attention was paid to using as standard a subset of COBOL as possible. In addition special routines were employed to allow the system to work at computer installations whose COBOL language does not allow as direct access to individual characters in its memory as it does to some larger units (i.e. indexing only on "words").

And finally, the Text Morphology Display System provides for linking morphs and their corresponding data together into word-like groupings under the control of the morph boundary punctuation. The following table illustrates the rules by which boundary punctuation controls word grouping.

		Current Morph Begins With		
		(	- + #	other
Previous Morph Ended With	)	Start New Word Group	Concatenate Morphs	Start New Word Group
	- + #	Concatenate Morphs	Concatenate Morphs	Concatenate Morphs
	other	Start New Word Group	Concatenate Morphs	Start New Word Group

The linguist may use + -# in any way he sees fit. They are simply connectors to the system. Space is considered to be a word delimiter, and parentheses as framing stems. If two stems are to be shown in the same word, the first stem may be given as (FIRSTSTEM)-.

The user should note that morph-initial punctuation will group together certain types of morphs alphabeticly within the class indicated by the initial punctuation.

James D. Ashworth
2005 Hyde Ave.
LaCrosse, Wi. 54601

The following page contains a system flow chart for the Text Morphology Display System. Such charts are intended to depict the flow of materials and information as well as the general processes they undergo in an information processing system.

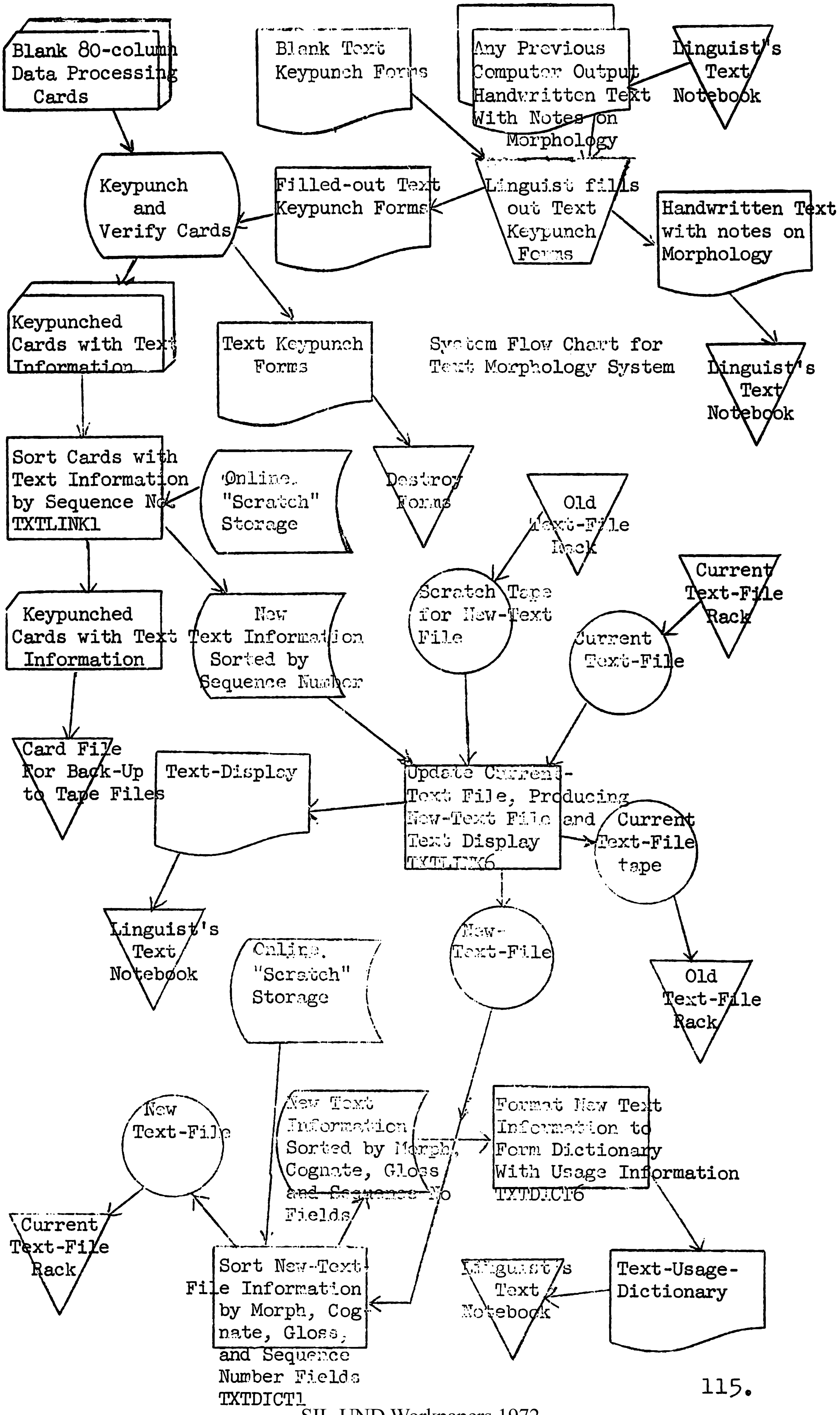
Processes are described within rectangles, trapezoids, and rectangles with rounded ends.

A triangle with downward apex is used to designate a place for long-term information storage.

The other closed figures designate media for information processing.

Directed lines show information and material flow from step to step.

"Scratch" as in "scratch tape" and "online scratch storage" indicates a computer-oriented storage medium that the system may use for temporary information storage without danger of destroying significant information previously recorded there.


Blank 80-column Data Processing Cards
Blank Text Keypunch Forms
Any Previous Computer Output Handwritten Text With Notes on Morphology
Linguist's Text Notebook
Keypunch and Verify Cards
Filled-out Text Keypunch Forms
Linguist fills out Text Keypunch Forms
Handwritten Text with notes on Morphology
Keypunched Cards with Text Information
Text Keypunch Forms
Linguist's Text Notebook
Sort Cards with Text Information by Sequence No. TXTLINK1
Online "Scratch" Storage
Destroy Forms
Old Text-File Rack
Keypunched Cards with Text Information
New Text Information Sorted by Sequence Number
Scratch Tape for New-Text File
Current Text-File Rack
Current Text-File
Card File for Back-Up to Tape Files
Text-Display
Update Current Text File, Producing New-Text File and Text Display TXTLINK6
Current Text-File tape
Linguist's Text Notebook
Online "Scratch" Storage
New-Text-File
Old Text-File Rack
New Text-File
New Text Information Sorted by Morph, Cognate, Gloss ~~and Sequence No~~ Fields
Format New Text Information to Form Dictionary With Usage Information TXTDICT6
Current Text-File Rack
Sort New-Text File Information by Morph, Cognate, Gloss, and Sequence Number Fields TXTDICT1
Linguist's Text Notebook
Text-Usage-Dictionary

System Flow Chart for Text Morphology System

Bibliography

Garvin, Paul L., 1962, "Computer Participation in Linguistic Research," in Language, v.38, pp. 385-9.

Grimes, Joseph E., 1963, Automatic Data Processing for You. A Guide to the Concordance Project. Norman, Okla.: Summer Institute of Linguistics and the University of Oklahoma Research Institute. 22 pp.

______, 1965, Review of Lamb and Gould 1964. IJAL 31:178-181.

______, 1967, Computer Support of Linguistic Field Work. Basic Concordance Program. mimeographed document available from University of Oklahoma Research Institute, Norman, Okla. 73069

Harris, Zellig S., 1962, String Analysis of Sentence Structure. The Hague, Mouton, 70 pp.

Lamb, Sidney M. and Laura Gould, 1964, Concordances from Computers. Berkeley and Los Angeles: Mechanolinguistics Project, University of Calif. 90 pp.

Samarin, William J., 1967, Field Linguistics. New York, et al, Holt, Rinehart and Winston, 246 pp.

Linguist		Page of
Language		Date
Title of Text		

SEQUENCE NUMBER	MORPH 9	COGNATE 25	GLOSS 41