DOCUMENT RESUME

ED 105 715 FL 006 234

AUTHOR	Wang, William S-Y; Chan, Stephen W.
TITLE	Development of Chinese-English Machine Translation System. Fnal Technical Report.
INSTITUTION	California Univ., Berkeley.
SPONS AGENCY	Rome Air Development Center, Griffiss AFB, N.Y.
REPORT NO	RADC-TR-74-22
PUB DATE	Feb 74
NOTE	153p.
EDRS PRICE	MF-$0.76 HC-$8.24 PLUS POSTAGE
DESCRIPTORS	*Chinese; *Computational Linguistics; *Contrastive Linguistics; Descriptive Linguistics; English; Generative Grammar; Grammar; Lexicography; Lexicology; *Machine Translation; *Programing; Sentence Structure; Structural Analysis; Syntax; Uncommonly Taught Languages

ABSTRACT

The report documents progress and results of a 2-1/3 year effort to further the prototype Chinese-English Machine Translation System. Additional rules were incorporated into the existing grammar for Chinese analysis and interlingual transfer, with emphasis on the latter. CHIDIC was updated and revised. Approximately 16,000 new entries were added to CHIDIC, bringing the total available entries to over 73,000. Linguistic work on a random access dictionary incorporating feature notation was carried out. A new design for the translation system was initiated and partially programmed for conversion of the current system from a CDC 6400 version into an IBM version. Better control of the parsing process was achieved by improving the segmentation procedures during input, and by addition of more revealing diagnostic printouts as steps toward reduction of spurious ambiguities. The Model 600D Chinese Teleprinter System was used for the first time to prepare large batches of texts for input. A total of 307 pages of machine readable texts, comprising 300,000 characters were prepared during this report. (Author)

RADC-TR-74-22
Final Technical Report
February 1974

DEVELOPMENT OF CHINESE-ENGLISH MACHINE TRANSLATION SYSTEM

The University of California at Berkeley

Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, New York

DEVELOPMENT OF CHINESE-ENGLISH MACHINE TRANSLATION SYSTEM

Dr. William S-Y Wang
Mr. Stephen W. Chan

The University of California at Berkeley

FOREWORD

This final report was prepared by The University of California at Berkeley under Contract F30602-71-C-0116, Program Element No. 62702F, Job Order No. 45940801. The RADC Project Engineer was Zbigniew L. Pankowicz (IRDT).

The authors were Dr. William S-Y Wang, Mr. Stephen W. Chan, Dr. Benjamin K. T'sou; the contributors were Mr. Stephen P. Baron, Mr. Harold Clumeck, Mr. Herbert Doughty, Mr. Cheung Fung, Mr. Robert Gaskins, Mr. John Hou, Mr. Robert Krones, Mr. Philip Robyn, Miss Susan Schultz, Mr. Ronald Sykora, and Miss Stella Ting.

This report has been reviewed by the Office of Information, RADC and approved for release to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved.

APPROVED: Zbigniew L. Pankowicz

ZBIGNIEW L. PANKOWICZ
Technical Evaluator

APPROVED: Howard Davis

HOWARD DAVIS
Technical Director
Intelligence and Reconnaissance Division

FOR THE COMMANDER: Carlo P. Crocetti

CARLO P. CROCETTI
Chief, Plans Office

TABLE OF CONTENTS

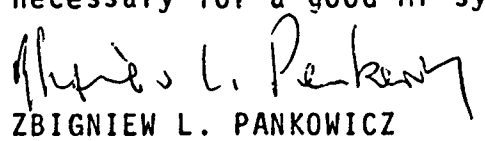
TECHNICAL EVALUATION

"Development of Chinese-English Machine Translation System"

The report documents a two year performance in Chinese-English MT R&D (1 September 70 - 30 August 72), including a programming effort oriented toward conversion of the system to IBM 360/65. The latter extended the lifetime of the contract through the end of December 72.

As indicated in Section I (INTRODUCTION), this development constitutes "a practical combination of the theoretical and the pragmatic approaches to machine translation." (p.2). The developmental strategy takes into account distinct peculiarities of Chinese and English in view of the tact that these two languages have no common lexical features and exhibit no general similiarity in structural aspects. The report points out in this respect that this high degree of dissimilarity effectively prevents exploitation of "the general resemblance of structure between source and target languages which has been relied upon in machine translation systems for pairs of European languages." (p.4). The vast amount of differences between Chinese and English prompted adoption of a contrastive approach to machine translation, as shown in the operation of GRAMMARS performing syntactic analysis of Chinese, quote, "The data base required here is a set of augmented context-free grammars designed to expose points of contrast with English." (p.6). The entire description of the system under development (Sections II-IV) dwells on dissimilarities between Chinese and English and concentrates heavily on the present and projected methods for their most judicious exploitation in the context of machine translation.

Section VIII (CONCLUSION) highlights improvements in syntactic analysis and the overall reduction of ambiguities as a direct result of a "careful research into the properties of Chinese" (p. 140). Further improvements are envisioned primarily on the semantic and pragmatic level of R&D. The report points out that "the work in artificial intelligence research appears to present a method for capturing the semantic and pragmatic information necessary for a good MT system." (p. 141).



ZBIGNIEW L. PANKOWICZ

ABSTRACT

This report documents progress and results of a two-year effort to further develop the prototype Chinese-English Machine Translation System. Additional rules were incorporated into the existing grammar for Chinese analysis and interlingual transfer, with emphasis on the latter. CHIDIC was updated and revised. Approximately 16,000 new entries were added to CHIDIC, bringing the total available entries to over 73,000. Linguistic work on a random access dictionary incorporating feature notation was carried out. A new design for the translation system was initiated and partially programmed for conversion of the current system from a CDC 6400 version into an IBM 360 version. Better control of the parsing process was achieved by improving the segmentation procedures during input, and by addition of more revealing diagnostic printouts as steps toward reduction of spurious ambiguities. The Model 600D Chinese Teleprinter System was used for the first time to prepare large batches of texts for input. A total of 307 pages of machine readable texts, comprising 300,000 characters were prepared during this period.

I. Introduction

The proto-type version of the Berkeley Chinese-English Machine Translation System, called the Syntactic Analysis System (SAS) was described in detail in our previous technical report under Contract No. 30602-69-C-0055. (Wang et al. 1971) The work in the period under report is a continuation of this effort. In both linguistics and programming, improvements to the translation system have been made. Advantage was taken of the requirement to convert the system currently operating on the CDC 6400 into a version operable on the IBM 360/65 for initial capability by further reorganization and redesign of the SAS to reflect newly acquired research results.

This report documents the further work on the analysis of Chinese, the interlingual transfer rules necessary for translation into English, additional components of the SAS which have been added and/or under further development, and the acquisition of a new character input system.

The orientation of the Berkeley project is a practical combination of the theoretical and the pragmatic approaches to machine translation. The techniques of current linguistic theory are used in any area where they have progressed far enough to offer effective results in translating Chinese, and the parts of the system which are most satisfactory at the moment are those which can take advantage of good theoretical

analysis. But, as is well known, there are many aspects of language for which current theory offers no analysis suitable for inclusion in a machine translation system, and in these areas the Berkeley group does not hesitate to incorporate heuristic procedures reflecting the best current practical knowledge of Chinese and English. In each area the approach which offers the best way of translating Chinese is adopted, within a general conceptual framework which must be clear enough to facilitate continuing development.

The aim of the Berkeley machine translation system is to accept a Chinese text exactly as printed, with no pre-editing of any kind, and to produce an English version of the same text in a form suitable for post-editing by human editors. The basic strategy of translation proceeds in two main phases. The first is "analysis," the phase of analyzing the Chinese text to recover from it as much information as possible about its structure. The analysis phase is free-ranging, and collects whatever facts about the Chinese which may be relevant to the translation task. The second phase is "synthesis," the phase of synthesizing an English output to correspond to the Chinese. The synthesis phase is target-directed, in the sense that there are many requirements on what is an acceptable English text, and the system tries to satisfy as many of these requirements as possible from the information gathered about the Chinese during analysis.

Although these two main phases of processing are obviously closely connected, the conceptual division into "analysis" followed by "synthesis" has proved fruitful, since it recognizes that the operations closest to Chinese input must be Chinese-oriented, the operations closest to English output must be English-oriented, and there must be a clear and explicit interface in the middle between the two languages. Such an organization is particularly necessary for the specific language pair of Chinese and English, since the two languages are so very different and there is not the general resemblance of structure between source and target languages which has been relied upon in machine translation systems for pairs of European languages.

This two-phase strategy could be used on translation units of any size, but the Berkeley system currently uses it on one Chinese sentence at a time, proceeding sentence by sentence, with only a limited amount of information retained from sentence to sentence as global context. This is a good practical choice, since it is not infrequently true that a single long complex Chinese sentence will naturally translate into a sequence of shorter English sentences.

The operation of the Berkeley system can be described under six headings, consisting of the process of Chinese character input followed by five general operations of the translation cycle. Each of these last five operations consists of a set of programs together with an associated linguistic data

base which provides the programs with whatever information they have about Chinese and/or English. A general sketch of each operation is given below to provide an overview of the system's operation. The basic operations were already described in detail in the Technical Report previously mentioned.

Following input of the Chinese character text, then, the first operation of the translation cycle is called SEGMENTS. This operation locates the next sentence of the Chinese text, and segments it into sub-sentences for processing on the basis of graphic clues in the input. The data base associated with this operation is a set of interpretation tables which provide information on the special functions of some characters and list characters which may occur in the text substituted for one or more other characters. The purpose of the SEGMENTS operation is to uncover the structure of the input string on the basis of graphic symbols alone, insofar as that can be done.

The next operation is called LEXICON, and is basically concerned with the identification of words in the sub-segments already identified, and with information about the word-level of the Chinese text. The data base employed is a large Chinese-English dictionary, organized by Chinese lexical items, and containing for each its grammatical coding, its English translation equivalent, and a great variety of linguistic information about the lexical items and possible contexts for their use.

Following this, the operation of GRAMMARS is applied to the results of LEXICON, to analyze the syntactic organization of the Chinese sentence. The data base required here is a set of augmented context-free grammars designed to expose points of contrast with English. The application of GRAMMARS is the last operation of the phase of "analysis," and its results are now used to begin the "synthesis" of the English output.

The next operation is called TRANSFER, and is the process of converting the analyzed Chinese structure into a corresponding English structure. It does this by carrying out a set of interlingual transfer specifications, relating Chinese and English structures.

The last operation, the final one of the "synthesis" phase and of the whole translation cycle as well, is called EXTRACT. The goal of EXTRACT is to produce the proper string of English words representing the structure which has been synthesized. Following the collapsing of the structure, EXTRACT consults its data base of facts about English words, their regularities and irregularities, and edits the output string to conform. When this process has been carried out the translation is complete, and the cycle begins anew with the next sentence of the Chinese input.

It should be noted that the particular language-pair of Chinese and English presents a great number of interesting and difficult problems, both theoretical and practical, for machine

translation. The Berkeley system incorporates good or adequate solutions to many of these problems, and has partial solutions to others, but there still remain areas which are a challenge to continuing research, on which only a beginning has been made. It is the gradual solution of these remaining difficulties which will permit the Berkeley system to improve in the years ahead, but the discussions here are focused on the better-understood problems studied under this contract and whose solutions should provide the basis for the initial capability for translating Chinese to English in the immediate future.

II. SEGMENTATION

When a text has been input, the first step of the translation process consists of segmenting it into appropriate units which the system can adequately handle. The process of locating "sentences" is not difficult in Chinese scientific or technical text because since the early years of this century Western-style punctuation--periods, semicolons, and the like--have been used in such texts. But the structure of those units which Chinese writers mark off with periods is somewhat different from what we are used to as a "sentence" in English--a natural translation of a stretch of Chinese between two periods is apt to be a sequence of English sentences. Accordingly, the process of segmentation has to be continued within a Chinese sentence unit to find sub-sentences, or what the Berkeley group also calls "parse-units" since eventually the sub-sentences will be the first candidates for syntactic analysis.

For this purpose the sentence is segmented into parse-units by taking note of such things as commas, parenthesized expressions, formulae, and some Chinese characters which have rather fixed syntactic functions in the language. (See Fig. 1)

This process is by no means as certain as is the division into sentences, and so the system has to be prepared to reverse its decisions at this level, splitting further or re-combining, as more is learned about the sentence during

此種彈性散射反應，常發生在中子質量較低及原子質量較輕（例如水）之情形。

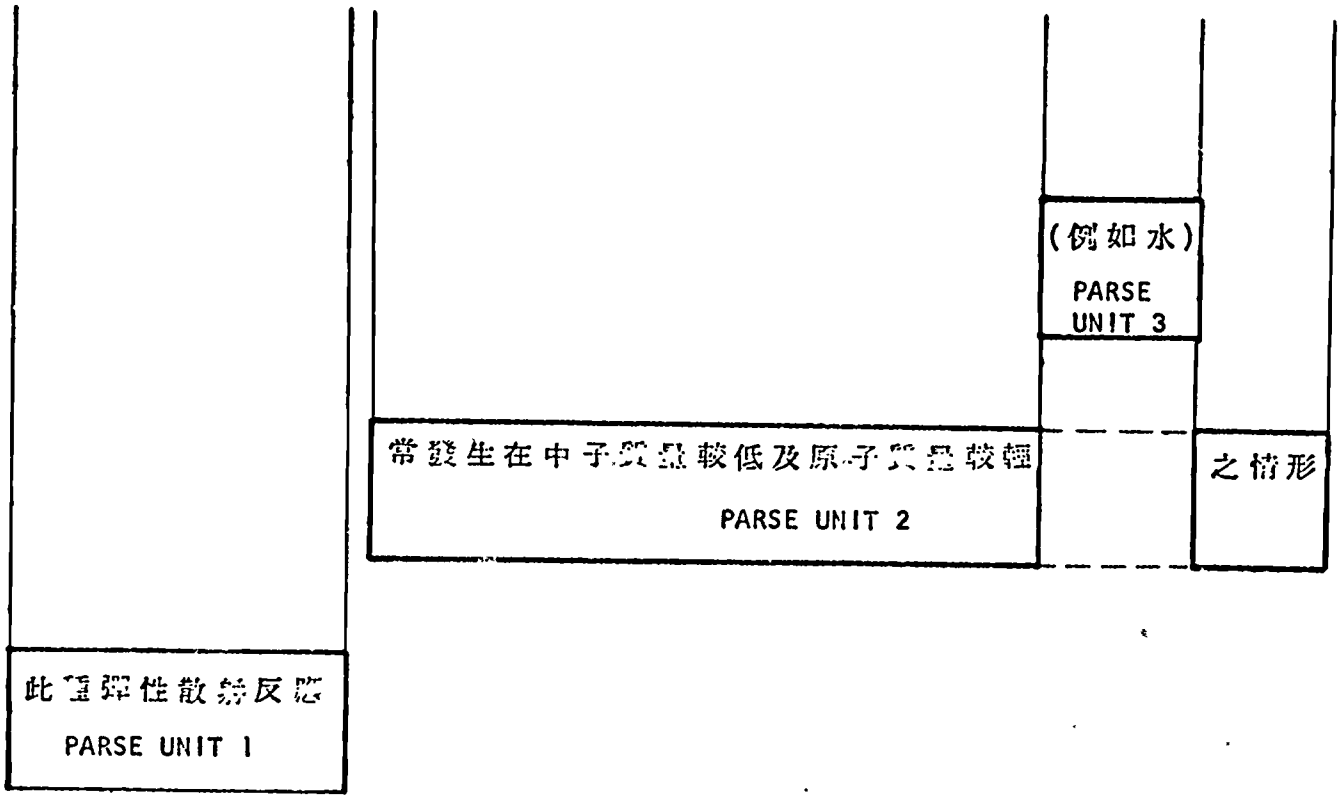


Figure 1. Segmentation of Sentence Into Parse Units

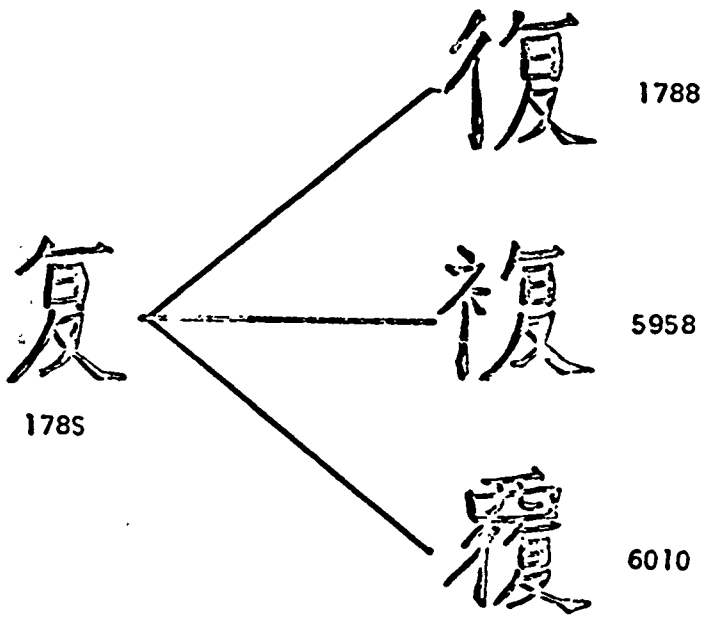
analysis.

Part of the reason why this sub-segmentation must be so tentative is that the punctuation in Chinese texts does not bear so simple a relation to the structure of sentences as does the punctuation in an English sentence. In Chinese, for example, two complete expressions which an English writer would feel must be separated by at least a semicolon, may stand together with no more than a comma between them; but on the other hand a bit further down the page a rather complicated subject may be separated from its verb by a comma, where in a corresponding English sentence no punctuation at all could possibly occur.

This use of punctuation is so difficult that some attempts have been made to simply strip all punctuation marks out of Chinese texts in despair of ever handling it reliably. The Berkeley group feels that the proper treatment of punctuation is as a suggestive guide for first analysis attempts, making use of whatever information may be present in the text but never relying on the presence or absence of punctuation. This means, however, that the sub-segmentation cues must be handled heuristically and tentatively if a system is to be adequate to the complex facts of Chinese.

Many further complications have to be introduced at this stage in order to handle real Chinese text. For instance sometimes a character appears in the input which may represent either of two (or sometimes more) telecodes in the data bases

我們的問題十分[复]雜。



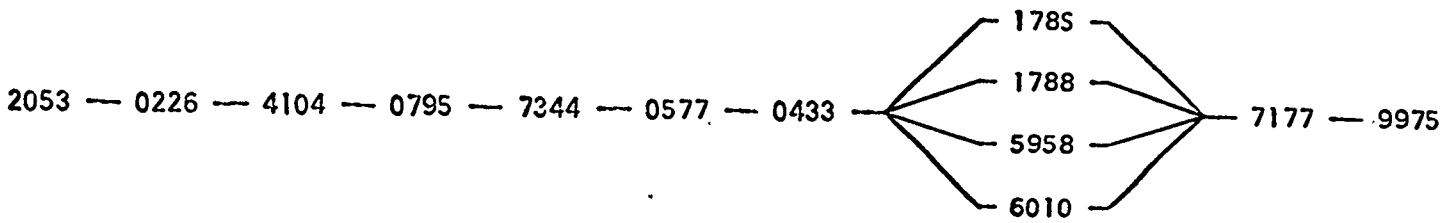


Figure 2. Multiple Telecode Substitution

used by the system. (See Fig. 2)

(One way in which this arises is from governmentally-sponsored reforms in the script, when a single character is introduced to represent what was formerly written as two or more distinct and different characters. The system must be able to handle texts written both before and after such a change, so the procedure of choice is to represent forms in lexicons using the distinguished characters; but then when the new single character is encountered there is no way of telling which of the distinct characters is intended, and so the possibility of both must be carried along. This gets as complicated as if some letters in an English text could not be deciphered, and so all the possible alternate spellings would have to be preserved until they could be looked up in the dictionary to see which could be real choices.)

Sometimes, too, it can be detected just from the string of characters that one character or more may have been elided or an expression shortened--in this case, the character or sequence may well be "conditionally" supplied for insertion in case it should be needed, and analogously characters are "conditionally" deleted subject to later checks.

Currently the system can segment on two basic linguistic levels, (a) the sentence level and (b) the subsentence level. As mentioned earlier, in Chinese texts, the "sentence" is any string ending with a period or its equivalent, such as a

question mark or exclamation mark or a semicolon (which is extremely rare).

The subsentence level segmentation symbol is chiefly represented by the comma. Other symbols are the different types of parenthesis, dash, special spacing, etc. Any unit which SAS segments for processing is called a "parse unit".

Quite powerful extralinguistic information can be culled from these segmentation cues which can lead to better recognition and parsing of each parse unit. For example, the information about new paragraphing is a cue that new information in the following paragraph of the text may be introduced. This in turn will affect the assignment of pronominal reference within that paragraph. This type of information lies in the area of discourse analysis, which is only beginning to be formally studied.

A more immediate result of such careful segmentation is that these cues can be taken as representing acoustic cues in the spoken process. Thus they can be considered highly effective methods of isolating the correct constituents in a character string. Since there is no such thing as explicit word boundary indicators (e.g. the blank in English) in printed Chinese texts, the careful preservation of such information is of the utmost importance. However, as was already mentioned previously in our report, punctuation is not a well-defined representation in written texts. Therefore, previous attemtps

to build this information into the grammar rules themselves resulted in partial failure. But the heuristic use of such cues will increase the power of rules written without explicit incorporation of punctuation signs.

The following is a list of punctuation and formatting codes (in telecode representation) which are used for our two basic segmentation levels.

DELETE ALONG WITH FOLLOWING TELECODE:

999B	text i.d.
999C	sub-text i.d.
999P	page

EXTRACT TEXT BRACKETED BY:

996F, 997F	open footnote, close footnote
9988, 9989	open parenthesis, close parenthesis
998B, 998C	open square bracket, close square bracket
999X, 999Y	begin formula, end formula

DELETE ALONG WITH TEXT BRACKETED BY:

999A	plot label information

SEGMENT JUST PRIOR TO AND DELETE:

999H	heading
9998 9998	space or blank (two in a row)
9990	dash (restore in English output)

SEGMENT JUST AFTER AND RETAIN:

99975*	period 1
9976*	comma 1
9979	semicolon
9980	colon
9981*	question mark
9982*	exclamation mark
9991*	Chinese ellipsis

* unless followed by 9985 (close single quote) or 9987 (close double quote), in which case segment after and retain 9985 or 9987 rather than the starred items above.

DELETE:

985S, 986S, 987S	superscirpt shifts
985T, 986T, 987T	subscript shifts
985B, 986B, 987B	boldface shifts
985I, 986I, 987I	italic shifts
985C, 986C, 987C	capital shifts
9999	new line

DELETE, OR CALL 'NAME' SUBGRAMMAR TO PARSE TEXT BRACKETED BY:

9994	begin special or proper name
9995	end special or proper name

DELETE, OR CALL 'BOOK TITLE' SUBGRAMMAR TO PARSE TEXT BRACKETED BY:

9996	begin book title
9997	end book title

DELETE, OR UNDERLINE IN ENGLISH OUTPUT THE GLOSS STRING CORRESPONDING TO CHINESE TEXT BRACKETED BY:

9992	begin emphasis
9993	end emphasis

III. THE LEXICON AND DICTIONARY LOOK-UP

III.1 The Look-up Process

The process of dictionary look-up in a bilingual machine dictionary, familiar as the heart of machine translation systems since the early 1950's, must be elaborated considerably for the processing of Chinese. Moreover, although dictionary look-up becomes very complicated for Chinese, still it cannot be such a central component as in machine translation systems for other languages; the task of identifying a word in a Chinese text, and when identified the task of determining its general grammatical function, present problems completely unknown in processing European languages.

Since a Chinese text consists of a sequence of characters, each of which corresponds generally to a single syllable of the spoken language, there is a popular superstition that Chinese is a language containing only one-syllable (that is, one-character) words. That is not the case, and in fact the notion of a "word"--consisting of one or several syllables--is much the same in Chinese as in English. The important difference is one of representation; in Chinese, the division of a text into words is not represented in writing at all. Some very approximate notion of the difficulties caused by this fact can be gained by considering how inconvenient it would be to work with an English text in which all the words had been run

together without blanks between them.

Thus, the process of identifying word boundaries must proceed by consulting the lexicon to see what words can be discovered. It is generally useful to look for longer items before considering shorter ones, but this strategy does not always give correct results and so it is necessary for some items in the dictionary to indicate other word segmentations which should be attempted. In difficult cases the Berkeley group has its system resort to looking up the text string in the lexicon once from left to right, and then a second time from right to left, ('Double Look-up') accepting the (sometimes quite different) words identified by both look-ups.

In many cases a definite decision about word boundaries cannot be arrived at without syntactic and semantic information. (Surrounding context of words, though sometimes useful, is of limited value since in any particular case the context itself may not be well-defined!) In these cases the alternative segmentations of the text into words must be accepted and carried along until a decision among them can be made later.

The other problem which Chinese presents in dictionary look-up arises once it has been decided that some string of characters should be identified as a word, for then the question comes up as to whether the word so identified is being used as a noun, a verb, an adjective or adverb, or just what its

grammatical category should be. In the European languages which have been the subject of machine translation research in the past this information can ordinarily be gained by inspecting the form of the word for "inflectional" and "derivational" markings, the special prefixes and suffixes which are attached to the stems of words to mark tense, number, gender, case, and the like, and from which the grammatical category of a word can often be deduced directly.

In Chinese, on the other hand, such explicit prefixes and suffixes do not appear. Consulting the lexicon about a word may tell you that it can be used as, say, either a noun or a verb, but will not be able to tell you which of the two uses you have in hand. Here again, decisions about grammatical functions of lexical items cannot in general be made without further syntactic and semantic information, and so multiple alternatives must be accepted from the lexicon and carried along for later decision, just as with the alternative word segmentations.

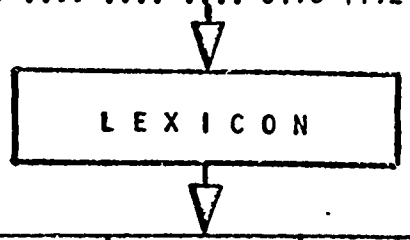
One implication of these facts for a machine translation system is that, since no word or grammatical function of a word can be definitely identified apart from the full range of alternatives known in the lexicon, it is not possible to use any variant of the familiar scheme of the small, high-frequency dictionary backed up by a much larger full dictionary residing on a slower class of storage. There is no substitute for

reference to the full lexicon all the time, which means that something roughly corresponding to an efficient disc-based information retrieval system must be programmed to do the look-up.

The unpleasant paradox is that for processing Chinese it is more important than for European languages that the dictionaries have very full coverage and be correct and complete in their linguistic information, that accessing them is more expensive, and that still the pay-off from them is less. Everyone familiar with machine translation knows that from the earliest work up to the present day, what are essentially very simple systems consisting of word-for-word substitutions as their basic translation strategy have often been able to give very plausible results. Such a system applied to Chinese produces near-total chaos, and the result of dictionary look-up for Chinese bears no relation to a word-for-word translation. (See Fig.3)

What is produced is a "sentence dictionary," the selection of items from the full lexicon which could be present in the sentence depending on how alternative segmentations and grammatical functions are resolved, and this collection is ordinarily two or three times the number of lexical items which will actually be determined later to be present. It is this collection which is passed along to the grammars for further analysis.

2974 4467 1734 1840 2414 1410 0646 .... 0173 1172 3055 9989 0037 1906 1748 9975



telecode string	grammatical category	English gloss		telecode string	grammatical category	English gloss
00221311	NN	NEUTRON		19061748	NBA	SITUATION
0037	DE1	'S		24141410	NB	SCATTERING
0037	DE2	OF		24141410	VIN	SCATTER
0037	DE3	WHICH/THAT		24141410	VTHB/NNS	SCATTER
0037	DE4	[no gloss]		2974	DD	THIS
0037	DE8	IN+WHICH		2974	NB5	THIS
0037	NRJ	IT/THEN		2974	IVG-NL	HERE
0144	VQ12	LOW		3055	DC	AQUO
01731172	CB	FOR+EXAMPLE,		3055	DP	AQUEOUS
06261311	DP	ATOMIC		3055	NN	WATER
06261311	NN	ATOM		40993932	NB	OCCURRENCE
0644	CN	AND		40993932	VI	OCCUR
06462019	NB	REACTION		40993932	VT	UNDERGO
06462019	VIN	REACT		4429	UN	KIND
0951	APR	BE+(VERB)-ING		4429	VTH/NA	PLANT
0951	VG11	BE+AT/BE+IN		63476852	DP	QUALITATIVE
0951	VG12	BE		63476852	NB	[PHYSICS]MASS
0951	VG13	IN/AT		63476852	NB	QUALITY
0951	VG14	[no gloss]		6525	AQ	COMPARATIVELY
1603	AA	OFTEN		6525	WQ	THAN
17341840	DP	ELASTIC		6535	VQ12	LIGHT
17341840	NBY	ELASTICITY				

Figure 3. Selection of Sentence Dictionary

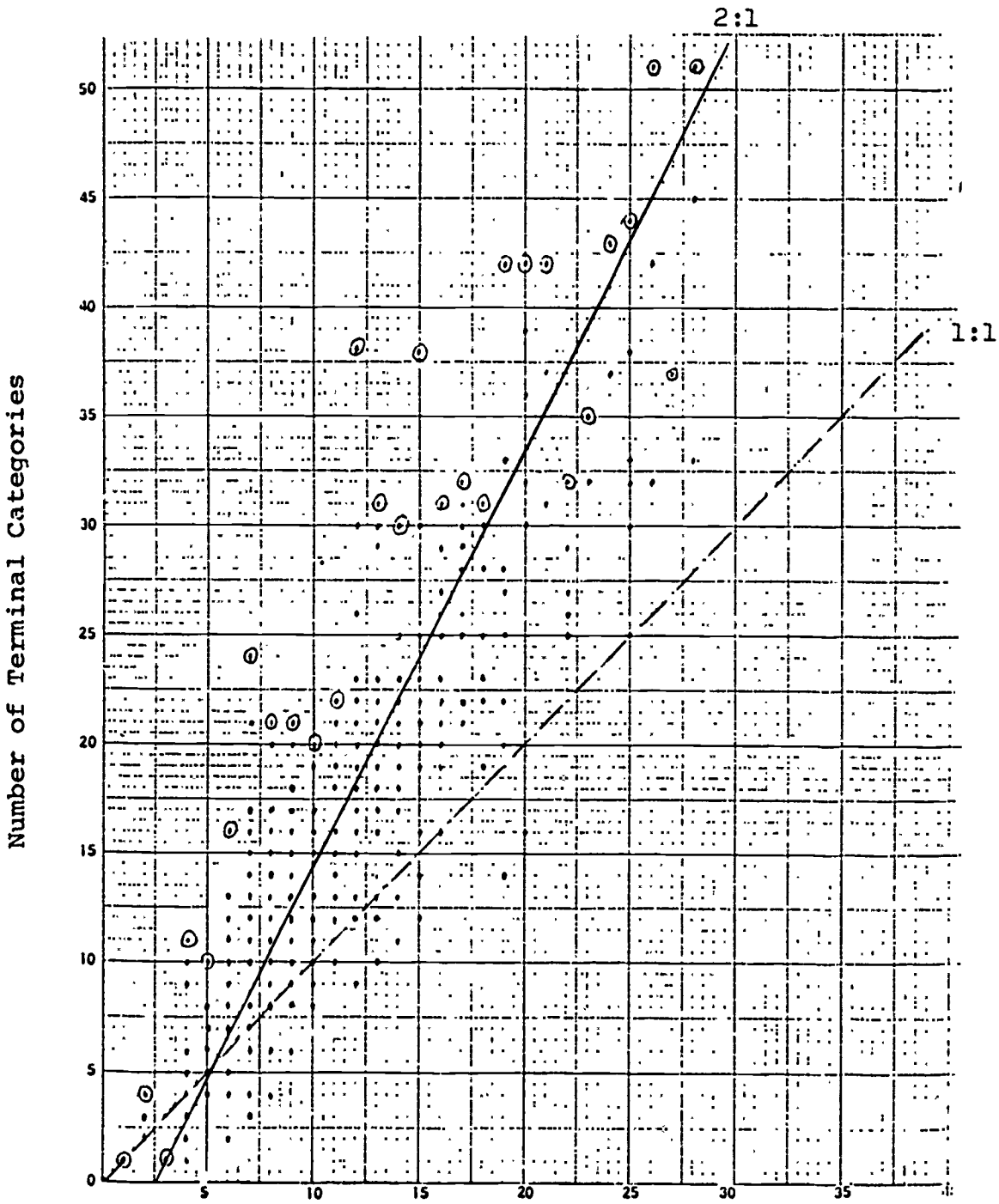

2:1
1:1
Number of Terminal Categories
Parse Unit Length (in characters)
0
5
10
15
20
25
30
35
40
45
50

Figure 4

An examination of the accompanying plot (Fig.4) representing the ratio of looked up terminals to the length of parse units is illuminating. This was taken from our Physics 5 Text which had 5,000 characters and 384 parse units with the majority of parse units having a length of between 10 and 20 characters. Consider the 1:1 ratio line through the graph implying roughly that there is one lexical entry for each character in the sentence. (This is equivalent to claiming that modern Chinese is a monosyllabic language). This is known not to be the case. Although no large scale data is available, it is safe to assume that for the general language, bisyllabic words, i.e. two-character words, is just as frequent. In a situation where every word in the sentence is correctly and uniquely looked up, the trend on the plot should show a line below the 1:1 ratio line. (i.e. fewer terminals versus length) Instead, we see a line which is above the 1:1 line (by least square fit). It is in fact a 2:1 ratio. The circled points, representing the maximum number of terminals for each sentence length shows a 3:1 ratio. An explanation for this comes from knowing the fact that there is minimal morphology in Chinese. Thus any one or two character word looked up would belong to two or three syntactic categories, as indicated by the number of terminals looked up versus length of the sentence. The task of 'disambiguating' this explosion in terminal categories has to be relegated to the syntactic rules and semantic feature checking components of the analytic process. The more complete

the dictionary, the more complex would be the results of dictionary look-up. It seems then that a system which relies heavily on dictionary look-up but not buttressed with sufficient syntactic and semantic rules would have a difficult time sifting through this mass of categories to arrive at the correct analysis of the Chinese sentence.

III.2 Revising the Lexicon

The lexicon and its data structure is so fundamental to the translation system that one cannot sufficiently emphasize the need for accurate encoding of information for each and every one of the items in the lexicon. In dealing with a large bilingual dictionary such as CHIDIC, which has accumulated over 73,000 lexical entries, the need to constantly update information requires extensive efforts in programming and in linguistic and lexicographic analysis.

Time and again, it has been our experience that a particular sentence would have been successfully parsed except for the fact that one item did not have the desired code. As a result the system tries other alternative parses and might come up with several results, none of which being the desired output. Our efforts during this period were devoted to a large scale revision of existing CHIDIC entries to ensure uniformity and accuracy in telecode representation, grammar code assignment and accurate English gloss equivalence which would

facilitate the output editing task either by machine or by man. The task was seen as a repetitive process, infusing more detail into the dictionary and systematizing its handling with each successive update. As aids in this task, several dictionary maintenance utility routines were written, and have benefited both the linguist and the lexicographer in contributing to their efficiency.

The revision work in lexicography emphasized the following areas:

1. Systematizing all Discipline Notation in existing CHIDIC entries.
2. Eliminating entries which will cause mismatch in look-up from left to right and/or right to left.
3. Redesigning the data structure of CHIDIC for Disk implementation.
4. Gradual implementation of Feature Notation.

(1) and (2) are continuing processes which were already begun in our preceding effort. Tasks (3) and (4) were begun during the present period and will continue into a following contractual effort.

III.3 New Data Structure for CHIDIC

Designs for a completely new data base for our dictionary were initiated. The data structure for this new

dictionary is quite different and much more flexible than the present format of CHIDIC. These new structures are now considered essential for an efficient utilization of the whole MT system under redevelopment. They grew directly out of our experience in our previous efforts in using the existing CHIDIC format.

In the existing system, CHIDIC is a sequential file consisting of the telecode entry, the associated grammar codes, English gloss and romanization of the telecode. For a particular run of text, it was necessary to select a sub-dictionary small enough to fit into the MT system. This sub-dictionary is now considered less desirable than using a full dictionary for the following reasons:

(1) Since it is necessary to update the dictionary every time new text is run, this should mean updating ideally both the subdictionary and CHIDIC at the same time. However, too frequent updating of CHIDIC is not economical when test runs are made. So the practical way has been to update the sub-dictionary frequently. This creates a sort of incompatibility in the time element between different versions of CHIDIC and versions of subdictionaries. The result is that linguists working on the latest rules and lexical entries sometimes get conflicting analyses due to uncorrected entries in CHIDIC itself.

(2) Another problem involves the uneconomical tasks of

keeping track of all slightly varying versions of these dictionaries by both the lexicographic and the programming staff. As many places as possible where human error may be introduced should be eliminated to streamline the processing task.

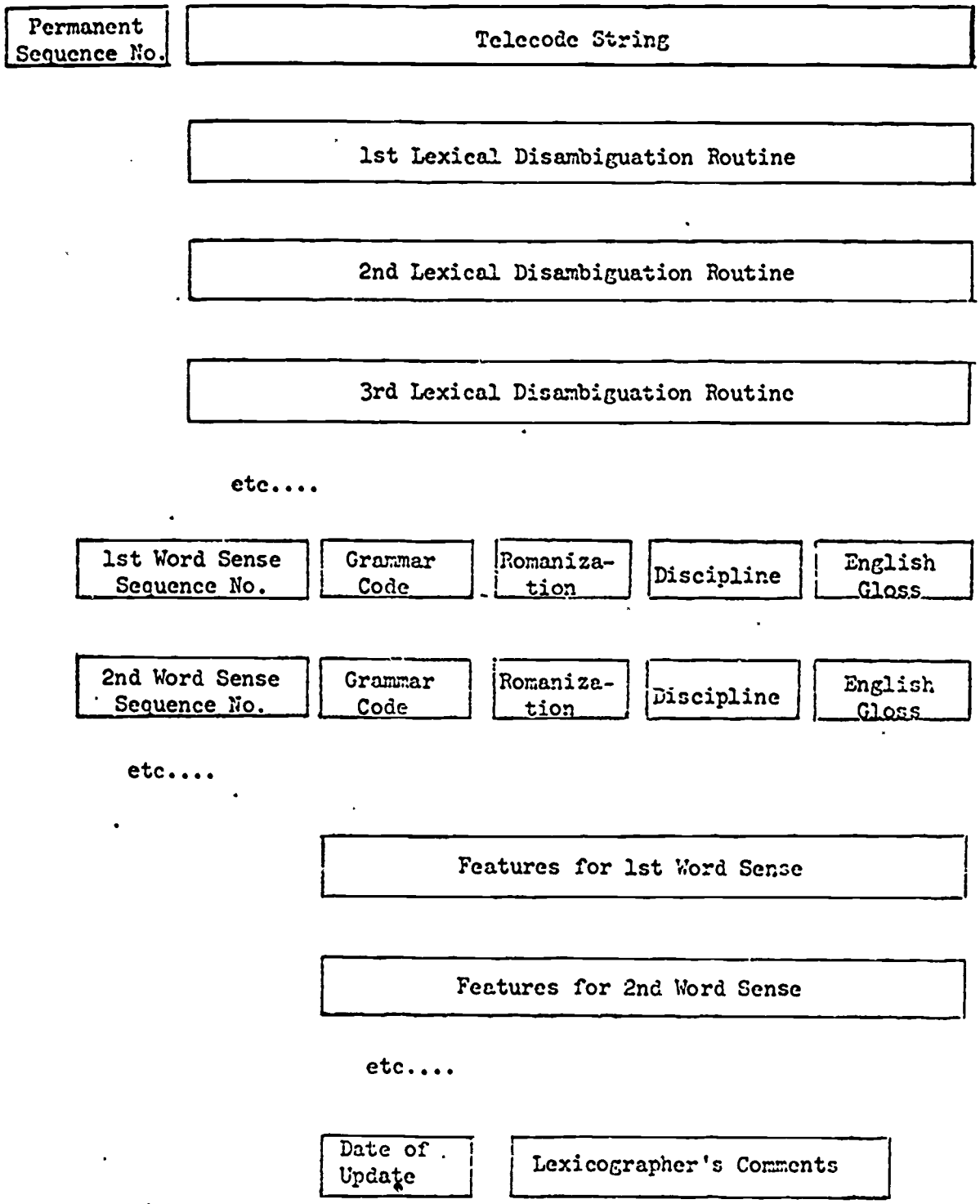
Our solution is then to make use of a full dictionary concept, in which it will be possible to update frequently and only once to one entire dictionary, but without assuming a burdensome cost of computer time. Since it is obvious that the whole dictionary cannot be resident in core, an economical way is to use the paging concept, where segments of the dictionary can be swapped in and out during lookup. The dictionary is to be stored on a random-access device such as the disk. We have come up with a more efficient method of search by using a "three-quarter telecode" search algorithm. It was found that instead of using all four digits of the telecode in searching and look-up, using three digits out of the four would make maximum use of available storage without sacrificing too much time.

Furthermore, every field of a dictionary entry will have to be capable of being separately updated. Each field will no longer be associated with a single output line where each field is fixed. In the updating process, our design is to allow for correction not only to individual fields in a particular entry, but to allow for the correction of even a single print

character which is found to be in error. This will increase the efficiency of the lexicographic staff in making corrections and other changes.

Since each subfield in an entry is capable of being accessed separately, this will make it possible to selectively process the information in each field. In particular, in the existing CHIDIC, it was rather difficult to manipulate the English gloss to reflect a better English output. The separate field for English gloss will make it easier of access.

Since the subfields will no longer be in fixed record format, they are now linked with pointers, allowing for a number of options on which combinations of subfields can be processed at any specific time. Schematically, a dictionary entry in disk CHIDIC will contain the following information:



Representation of Disk CHIDIC Entry

This new structure may be contrasted with our existing, much simpler dictionary format on CHIDIC:

Grammar Code	Telecode Entry	Romanization	English Gloss

The first difference is the fixed field format of this representation, which makes very stringent demands on the length of the telecode string (and consequently the length of the romanization string) as well as the gloss. Next is the lack of any information which will help to narrow down the number of word senses which the look-up program will submit to the parser for processing. We refer to this information under the general heading of lexical heuristics or lexical disambiguation routines. These are small routines which may be invoked singly or in Boolean combinations to arrive at the correct or most likely choice for a particular looked up entry. The structure of the dictionary is such that it will be possible to add or delete such routines as the state of research progresses. It is expected that concordances on selected entries of highest linguistic interest will be one of the best computational aids in arriving at some lexical heuristics which are dependent on distributional characteristics.

III.4 Features

Early in this contract the Project initiated the analysis of syntactic and semantic features of Chinese and English. We have continued to refine our ideas on incorporating features into our Syntactic Analysis System. It was noted that the grammar codes of the existing grammar already contain copious information regarding each syntactic subtype. For example, the class of nouns are already encoded with information stating whether a particular noun in the dictionary is animate or inanimate, human or '-human', abstract or concrete and even to the extent that certain nouns are parts of the body, etc. The same is true for other types of syntactic and semantic category of verbs and to a lesser extent adverbs, adjectives. All this information is capturable in terms of a system of features.

The incorporation of a system of features into our system involves the addition of a much more complex data structure to the dictionary. However, a systematic treatment of features will pay dividends in the simpler formation of grammar rules. In order to preserve the continual operating efficiency of our grammar, and to ensure that the transition be a smooth one, our approach has been to "translate" the information available in the grammar into feature codes, while at the same time completely preserving the form of our present grammar rules.

Our first step was to prepare to extract by machine the most obvious syntactic features from the grammar codes and then assign them a more systematic coding. This required preliminary work and was a necessity as some of our previous codes did not distinguish between syntactic and semantic feature information always in the same way and as a result the complexity of parts of the grammar was increased.

For example, the class of grammar codes beginning with the letter D were generally used to indicate the class of determiners or prenominal modifiers. The second character following D should then be subclasses, as is generally true for our present grammar codes. The second letter is usually either mnemonic or just follows the alphabetic sequence. So that subclasses of D are DA, DB, DC, DD, etc. However, there were also code sequences such as DA, DASH, DC, DD, DE, DF, DFS. "DASH" is the grammar code which is assigned to the graphic symbol '--' ('dash'). This already is one step away from systematic assignment, since one would prefer to group all punctuation-related symbols into a special class of codes, such as 'P', for the first letter, where already in fact 'period' signifying the end of a sentence does have the unique grammar code P. The next code to be considered is 'DE'. It is the grammar code for our well-known lexeme de (的), which does not come very well under a class of determiners nor adjectives. In the case of DASH, all four letters are purely mnemonic and together carry only one unit of information. As for DE, the two letters are

again mnemonic only and again together carry one unit of information. Whereas, for the majority of codes in this class each letter in sequence carries a unit of information. Similar inconsistency in coding is also evidenced for the codes beginning with the letter A (generally mnemonic for 'adverbs'). However, we have found intrusions such as 'ACUTEA' (for 'acute accent') and ASTERISK (for 'asterisk'), BRA (for 'open bracket'), UNBRA (for 'close bracket'). The first letter 'C' for conjunctions also included codes such as "COLON", "COMMA", etc. Inconsistencies of this type arise in various parts of the coding.

Another type of mixed coding also existed with regard to the second letter in terms of non-distinction between syntactic and semantic information. Consider the case of the types of nouns such as NA, NB, NC, ND, NH, NK, NL, NN, NT, NY, NZ. Syntactically, on the basis of current studies of Chinese structure, one can distinguish between four categories: concrete nouns, abstract nouns, time nouns and locatives. The grammar codes have NB for abstract nouns, NL for locatives and NT for time nouns. All the other codes named above should rightly belong to the concrete noun class and be indicated as such. However, it was only by implication that these other categories then should have the feature 'concrete'. The second letter of these codes actually provide various types of information such as

A = animate

C = chemical name

D = disease name

H = human

K = kinship

N = inanimate

Y = body parts

Z = chemical compound

Since some of these codes actually cross-reference others, (e.g. 'kinship' is a subset of 'human', which in turn is a subset of 'animate') it was therefore necessary to reassign symbols for the systematic extraction of features by our programs.

Resystematization was carried out during this contractual period. However, since the current SAS would not be able to absorb the greater complexity of these codes, the reassignment task was carried out separately and not directly incorporated into the existing CHIDIC coding. This will be done when the new system, designed to incorporate feature handling capabilities, is in operation.

The first steps in re-systematizing was carried out by going through all the grammar codes and assigning distinctive first letters to all existing categories, keeping as close to the present system as possible. E.g. the codes such as SEN (sentence), IND (clause), IND + BE - EN (Passive clause), INT

(interrogative clause), INS (subordinate clause), etc., have the characteristic first letter 'S' to indicate their membership in the category of sentences or clauses. A new first letter code P is now assigned to indicate the class of punctuation marks. Therefore, COLON, COMMA, BRA, DASH, SLASH, HYPHEN, QUEST ('question mark'), PERIOD now are consistently reclassified under 'P' so that they are no longer scattered throughout the alphabetic sequence.

The following examples exhibit some of the recoding of CHIDIC grammar codes by representing in a more explicit form information already available in each grammar code and its syntactic relations with other constituents as a result of examining the environments provided by the grammar rules.

	CHIDIC grammar Code	Function	Expanded Feature Coding	Remarks
1.	CC	clause conjunction	*,C,/LC,SID,/RC,SID,/ (e.g. huo 或 'or'	both left & right constitutents are clauses
2.	CN	noun conjunction	*,C,/LC,N,/RC,N,/ (e.g. ji 及, yiji 以及 'and')	both left & right constituents are nouns

	CHIDIC grammar Code	Function	Expanded Feature Coding	Remarks
3.	CS	subordinate clause conjunction	*,C,/RC,SID,/ (e.g. jiaru 假如 'if', suiran 雖然 'although')	right constituent must be a clause; no restriction on left constituent specifiable
4.	NH	human noun	*,N,+H/ (e.g. gongchengshi 工程師 'engineer'	
5.	NN	concrete noun	*,N,+PH/ (e.g. dahe 大核 'macronucleus', zidan 子彈 'bullet')	
6.	NN2	second level complex concrete noun	*,N,+2+PH/	
7.	NN2*R	" "	*,N,+2+PH+SR/	interlingual operation on NN2 required

	CHIDIC grammar Code	Function	Expanded Feature Coding	Remarks
8.	VTB/NA	transitive verb	*,VT/O,N,+BI+SP (e.g. zhong 種 'to plant')	requires object which is 'biotic' and 'self-propelling'
9.	VTH/NHS	transitive verb	*,VT,/S,N,+H/O,N,+H+PL/ (e.g. jieshau 介紹 'introduce')	human subject nonhuman object plural

A partial list of codes used in this feature implementation is given below:

LABELS

*	this node
S	subject of this node
O	Object of this node
V	verb modified by this node
N	noun modified by this node
SV	subject of verb modified by this node
A	adverb modifying this node
D	adjective modifying this node
C	complement of this node
LC	left of conjoined constituent

RC right of conjoined constituent

SID clause

FEATURES

H human

AH anthropomorphic

SP self-propelling

BI biotic

PO potent

PH physical (i.e., has mass)

TH thing (object)

QU quantizable

MA mass noun

TP time (point)

TD time (duration)

L locative

DS distance

DR direction

UN unique

PR proper

CH chemical

DI disease

BP body part

PL plural

The labels refer to the functional relations of each constituent with reference to a particular node in the tree. Thus for the

terminal categories such as CC given above, an asterisk * indicates that the same node as itself is referenced. The C following the * is the new code for the general category of conjunction. The constituent to the left of CC (labeled LC) has to be a clause (SID), and similarly, the constituent to the right (RC) also has to be a clause. There are no semantic features which need be isolated for CC or its left and right constituents. Whereas for VTH/NHS the feature representation says that the subject S of this transitive verb VT has to be a noun (N) with the feature human (+H) and the object (O) of this VT has to be a noun with the features human (+H) and plurality (+PL).

The advantages of such a resystematization are even more obvious in the next stage of our task. This consists of examining our grammar rules for consistency and completeness. For example, by merely abstracting the first letter of each grammar code from each rule, we were able to obtain a schematic shape of our present grammar. The linguist would have a clearer grasp of the form of the grammar without at all times being obscured by the very detailed subcategories of each major category. For example, when the linguist wishes to examine the grammar for rules that directly bring about sentential structures (i.e. the highest nodes in the resultant trees for any particular analysis), he has only to consult in the class of rules having 'S' as the first character code. This would comprise full sentences, clauses, subordinate and coordinate

structures, and even interrogative sentences. Suppose the linguist wishes to examine the rules represented by the schema:

(I) S --> N + V

(sentence) (noun phrase) (verb phrase)

There are in our grammar about 200 rules satisfying this schema, e.g.

(a) (1) IND -> NAS + VIASS

(2) IND -> NAS + VI3

(3) IND -> NN5 + VI3

(4) IND -> N + VQ

. . . . . .

(b) (1) IND + BE-EN -> NAS + VTH3

(2) INS + BE-EN -> NAS + VTHC

. . . . . .

(c) (1) INS -> NA5 + VIC

. . . . . .

(d) (1) INF -> NA5 + VIHAT

(e) (1) IND2 -> N + VIYE

(f) (1) SVT -> NXT + VTA3

(g) (1) SVU -> NXS + VXU

. . . . . .

It is clear that if we supply subscripts to rule schema (I) above, each of the rules (a) through (g) can equally be represented as

$$S_1 \rightarrow N_1 + V_1$$

$$S_2 \rightarrow N_2 + V_2$$

. . . . . .

$$S_{11} \rightarrow N_{11} + V_{11}$$

Going one step further in subclassification of sentence types we can again represent these as

$$S_{a1} \rightarrow N_{a1} + V_{a1}$$

$$S_{a2} \rightarrow N_{a2} + V_{a2}$$

. . . . . .

$$S_{b1} \rightarrow N_{b1} + V_{b1}$$

. . . . . .

$$S_{g1} \rightarrow N_{g1} + V_{g1}$$

It is the information in these subcripts that we have to capture in our rules. In fact, a vast amount of syntactic and semantic information is already captured in the present grammar codes, as was mentioned earlier. The concept of using feature matrices to represent this information now becomes much easier to implement. We can now systematically associate some set of features corresponding to the subscripts. (There is of course no implication of one-one correspondence between one feature and one subscript.

III.5 Parsing Incorporating Features

During the parsing process features of one word may be checked against its co-occurrence with another word within the same sentence. If the features of the words concerned are compatible, then the parsing goes forward and the grammar rules will be applied. If the features are incompatible, the rules will be blocked, thus eliminating certain illegitimate parses which might otherwise contribute to the ambiguity of the later analysis.

When the feature parser forms a new constitute from left and right candidate constitutes, it not only verifies that there is a rule in the grammar which assigns the category symbol of the new constitute to the concatenation of the category symbols of the candidates, but it also checks the compatibility of the semantic features of the candidates with the hypothesis that the candidates stand in the correct relationship to each other propounded by that rule.

In addition to its category symbol and the other fields which appear in an old style constitute, each new constitute will have relationship label fields for its left and right immediate constituents, and will also have a feature complex.

The feature complex of a constitute is an N-tuple of labeled feature matrices. The label of a feature matrix tells the relationship of the thing represented by that matrix to the

constitute in which the label occurs.

A feature matrix is a 4-tuple of feature vectors. The first vector of a matrix represents the features marked plus. The second vector represents the features marked minus. The third represents the features marked blocked, and the fourth represents the features marked overrideable. When a new constitute is made, its feature complex is built from the feature complexes of the left and right candidates in accordance with the parsing actions and labels in the current production.

If the resulting feature complex has self-contradictory markings, the formation of the new constitute is aborted.

The above regular and well motivated feature processing is augmented by the qualification process, by means of which a rule may make any ad hoc requirements on the feature complexes of the candidates and the resulting constitute.

Attached to each rule is an N-tuple of qualification alternatives, representing upper and lower bounds required of the left and right candidate feature complexes and a complex to be merged into the complex of the resulting new constitute.

It is anticipated that most rules will have vacuous qualification, that is, no ad hoc requirements, and that the rules with qualification alternatives will each make use of only a small part of the full power available for ad hoc specifica-

tion. These ad hoc specifications indicate that certain entries in the dictionary do not follow the general rules of constituent formation and must be treated by calls to special subroutines to effect a correct parse.

The incorporation of feature parsing into the new system is an incremental task, since its power is dependent on CHIDIC entries being fully specified with features. The initial capability of the system will be tested first using a smaller set of feature specifications which could be converted directly from the present CHIDIC grammar codes. As more entries become more fully specified in our new disk CHIDIC, we expect certain ambiguities which cannot be handled properly by the current grammar, such as that of noun compounding, will be more adequately resolved.

III.6 Supplemental Dictionary Sources

Besides obtaining new dictionary entries from regular bilingual technical dictionaries, the Project has extensively accessed entries in the FTD Nuclear Physics Dictionary. This is a very convenient source since entries were already in telecode also accompanied by Chinese characters and the English gloss. However, this dictionary was compiled for human translation and therefore the grammatical information must be supplied by us for each entry.

Another large source of technical terminology is the recently completed Technical Dictionary compiled by the Department of Defense. This is already on tape and distributed by CETA. We acquired the tapes near the end of the reporting period and have not yet had an opportunity to evaluate in detail its merit vis-a-vis our MT system. But the dictionary appears to have potential advantages.

IV. LINGUISTIC ANALYSIS AND INTERLINGUAL TRANSFER

During the period of this contract, the grammar rules were revised and expanded with special attention towards the interlingual mapping of Chinese structure onto English structure. Wherever possible such interlingual mapping will take advantage of the parallel structures that exist in the two languages and perform direct mappings instead of going through complicated analytic procedures, first in the Chinese sentence and then remapping these to English. (See also previous Technical Report Chapter VI) For example, if a noun compounding process in Chinese, say $N_1+N_2+N_3$ always has the same surface order $N_1+N_2+N_3$ in English, then efficiency can be increased by not making it necessary to analyse all the possible modificational structure of N_1, N_2 and N_3 such as $(N_1+N_2) + N_3$ or $N_1 + (N_2+N_3)$ or $(N_1) + (N_2) + (N_3)$. However, there will be cases where such deeper analysis is necessary when a Chinese compound of the form $(N_1+N_2) + N_3$ would have to be mapped into English as $N_3 + (N_1+N_2)$. The factors involved are quite complex, dealing with many subcategories of nouns and their semantic content, and work in this area has only scratched the surface. The analysis of lexical items into their syntactic and semantic features will be a step in the right direction. Recent studies such as those of Lees (1970), Zimmer (1971), Brekle (1970) have increased our understanding of English compounding but the Chinese case still has to be tackled. Li (1971) has made some

headway in this direction for Chinese.

The following sections will discuss selected areas of Chinese syntactic structure with reference to English contrastive structure and where appropriate rules were formulated or revised.

IV.1 Conjunctions

IV.1.1 Conjunctions for Clauses

At present, CHIDIC contains the following terminal codes for conjunctions:

CB - "exemplifiers", e.g. 'liru 例如 X' "for example, X"

CC - "disjunctions", e.g.

'haishi 還是 ' - "exclusive or"
'huozhe 或者 ' - "inclusive or"

CM - "numerical conjunctions", e.g. 'cheng 乘 ' as in

'X cheng 乘 Y' "X times Y"

CN - "nominal conjunctions", e.g. 'gen 跟 ', 'yu 與 ', 'he 和 ' as in

'X gen 跟 Y'
'X yu 與 Y' "X and Y"

'X he 和 Y'

CP - "paired conjunctions", e.g. 'budan 不但 ', 'erqie 而且 ', as in

'budan 不但 X erqie 而且 Y'

"not only X but Y"

CS - "subordinating conjunctions", e.g. 'yinwei 因為 ', 'yaoshi 要是 ', 'jiran 既然 ', 'suiran 雖然 ', as in

{'yinwei 因為 / 'jiran 既然 / 'suiran 雖然 / 'yaoshi 要是} 他來了 ta lai le'

"because / "since / "although / "if — he came"

CV - "verb conjunctions", e.g. 'er 而 ', 'he 和 ' as in

動物不能 恢復而 死亡
'Dongwu buneng huifu er siwang.'

"Animal could not recover and died."

CI - "sentential conjunctions", e.g. 'suoyi 所以', 'raner 然而', 'keshi 可是' as in

'suoyi 所以	
'raner 然而	他沒有來 ta meiyou lai'
'keshi 可是	
"therefore	
"however	he didn't come"
"but	

Special attention has been directed towards rules for conjunction types CI, CS, and CP. The first revision involved the CP category, for which the only rule extant was

(1) INS -> CP + IND

where IND is roughly anything that can act as an indicative expression, and INS is a subordinate clause. Among the many reasons why this rule is inadequate are:

(a) The sequence CP+IND+CP+IND as in

(2) YI fangmian women buneng zou; (IND)

ling yi fangmian (CP) women liu zai zhe-er (IND)

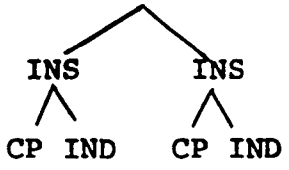
geng weixian. (IND)

一方面我們不能走；另一方面我們留在這兒更危險.

On the one hand we cannot go; on the other hand we stay at here more dangerous.

"On the one hand, we cannot go; on the other hand, it is even more dangerous for us to stay here."

(where 'yi fangmian 一方面' and 'lingyi fangmian 另一方面' are CP's) will get parsed as:



This means that in using rule (1) we would be forced to derive sentence (2) from a concatenation of subordinate clauses, a rather undesirable solution.

(b) Contrary to rule (1), CP need not be followed by IND, but can also be followed by a simple predicate, as where a subject noun phrase has been transposed to before the CP, or deleted.

The following sentence illustrates both cases:

(3) Ta budan hui shuo yingwen, erqie hui shuo jungguohua.
CP Predicate CP Predicate

他不但會說英文，而且會說中國話.

He not only can speak English, furthermore can speak Chinese.
"Not only can he speak English, but also Chinese."

A solution to the above problems is suggested if we note that pairing is often optional when a so-called CP precedes a predicate or IND, so that some sentences may actually contain only one "CP":

(4) Ta budan (CP) hui shuo yingwen, erqie (CP) ta hui shuo jungguohua.
他不但會說英文,而且他會說中國話.
He not only can speak English, furthermore he can speak Chinese.
"Not only can he speak English, he can also speak Chinese."

(5) Erqie ta hui shuo jungguohua.
而且他會說中國話.
Furthermore he can speak Chinese.
"He can also speak Chinese."

(6) Ta budan hui shuo yingwen, erqie hui shuo jungguohua.
他不但會說英文,而且會說中國話.
He not only can speak English, furthermore

can speak Chinese.

"Not only can he speak English, but also Chinese."

(7) Erqie hui shuo yingwen.

而且會說英文。

Furthermore can speak English.

" He can also speak English."

(8) *Ta budan hui shuo yingwen.

*他不但會說英文。

*He not only can speak English.

*"Not only can he speak English."

(9) Ling yi fangmian women liu zai zhe-er geng weixian.

另一方面我們留在這兒更危險。

On the other hand we stay at here more dangerous.

"On the other hand, it is even more dangerous for us to stay here."

In the above examples, it can be seen that 'budan 不但 ' acts much like a subordinating conjunction, whereas 'erqie 而且 ', 'yi fangmian 一方面 ', etc. act much like sentential conjunctions. This means that we should be able to use rules parsing strings with CS and CI to parse strings such as (2) - (9) containing 'budan 不但 ' and erqie 而且 '

type conjunctions, if these last are added or transferred to the appropriate category (CS or CI) in CHIDIC. Note for instance, that since subject NP transposal occurs before CS and is already covered by CS rules, subject NP transposal before 'budan 不但 ' as in sentence (3) will now be taken care of.

The unique case where pairing is obligatory is with strings of the type CP+N+CP+N, where N is a noun phrase, as in:

(10) Budan Zhang San erqie Li Si dou qule.
不但張三而且李四都去了.
Not only Zhang San but also Li Si all go (past).
"Not only Zhang San but also Li Si both went."

Likewise, the only rules involving CP will be of the form N→CP+N+CP+N. Finally, the only conjunctions that can participate in constructions such as (10) are 'budan 不但 ' and its synonyms, and 'erqie 而且 ', which, since they participate in other constructions as well (cf. sentences (2) - (9)), will now be listed in CHIDIC as follows:

CP 'budan 不但 ', etc.
CP 'erqie 而且 '
CI 'erqie 而且 '
⋮
CS 'budan 不但 '
⋮

Of course, the rest of the CP's, such as 'yi fangmian 一方面', etc. would have to be completely redistributed into the CI and CS categories, and will no longer appear under CP in CHIDIC.

Among the many problems which require further research are:

(a) Interlingual Disambiguation/Deletion. Several conjunction sequences will have to undergo disambiguation/deletion as part of the Chinese to English interlingual transformations. Note the following examples:

(1) Yinwei wo meiyou lai, suoyi ta ye meiyou lai.
因為我沒有來,所以他也沒有來.
Because I did not come, therefore he also did not come.
"Because I did not come, he did not come either."

(2) Wo suiran meiyou lai, keshi ta lai le.
我雖然沒有來,可是他來了.
I although did not come, however he come (past).
"Although I did not come, he came."

(3) Yaoshi ta lai de hua, wo jiu bulai.
要是他來的話,我就不來.

If he come if, I then not come.

"If he comes, then I will not come."

(4) Ta lai de hua, wo jiu bulai.
他來的話,我就不來。
He come if, I then not come.
"If he comes, then I will not come."

(5) Yaoshi ta lai, wo jiu bulai.
要是他來,我就不來。
If he come, I then not come.
"If he comes, then I will not come."

In sentences (1) and (2) we have what might be called "subordinate-coordinate" sequencing of conjunctions. In English this sequencing is more restricted than in Chinese; "because" may be followed by "therefore", but "although" may not be followed by "but". Therefore in the interlingual component there will have to be a rule deleting the gloss for keshi 可是 'but' when (and only when) a preceding clause contains suiran 雖然, as in sentence (2) above. Sentences (3-5) illustrate various ways of expressing the conditional conjunction in Chinese. Note that the sequence de hua 的話 is roughly equivalent to English 'say' as in "if, say, X does Y". One way to handle these sentences is to always translate de hua 的話 as 'say', at the same time shifting it to its correct position. This will give "Say he comes,...." in (4), which may be confusing to some speakers of English. Another solution

might be to delete de hua 的話 in every case and add "if" if yaoshi 要是 does not appear.

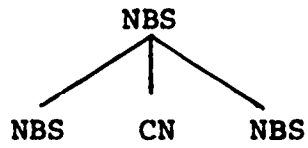
(Note: '... 的話 ' would perhaps be better translated as 'IF + IT + BE + THE + CASE + THAT' or simply "IF", rather than 'say'.)

IV.1.2 Conjunctions for Nouns

These involve constructions having the nominal conjunctions (grammar code CN) as part of their structure.

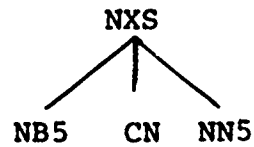
Suppose a certain string, after dictionary lookup, contains the sequence N_1+CN+N_2 as a substring. Existing rules indicate, and quite correctly, that one of the major properties of the category CN is that it conjoins two nouns (or noun phrases) having rather similar properties. Thus they are rules having the following structures:

(1)



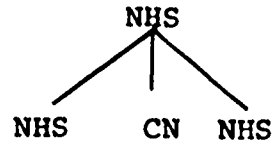
(abstract complex plural noun phrases)

(2)



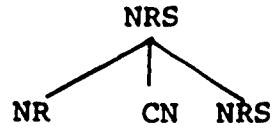
(complex noun phrase)

(3)



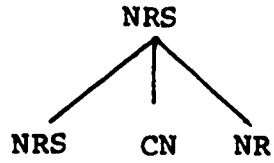
(human noun phrases)

(4)



(pronominal noun phrases: singular conjoined with plural)

(5)



(pronominal noun phrases: plural conjoined with singular)

Extrapolating from this type of structure, it can be seen that occurrences of "slightly different" combinations of conjoined noun phrases must be represented eventually by an exhaustive list of rules which will represent every allowable occurrence of such noun sequences. Doing this directly would mean easily adding a few hundred rules to the present grammar, but without essentially increasing its efficiency. As a matter of fact, practical considerations of computer storage would discourage such a brute force method of analysis and implementation. For example, the following string would present two ambiguous readings:

Congdon	he	Lorentz	de	shiyan
Congdon	and	Lorentz	DE	experiment
NM	CN	NM	DE	NB

which can have the translation of either

(a) Congdon's and Lorentz's experiment

i.e. the experiment of (both) Congdon and Lorentz

where the bracketing would be

(NM CN NM) DE NB

or

(b) Congdon and Lorentz's experiment

(i.e. only Lorentz's experiment was involved)

where the bracketing would be

NM CN (NM DE NB)

Theoretically, these involve problems of Phrasal Conjunction, which have been discussed in recent linguistic literature (Lakoff & Peters 1969) and have not yet received any concrete resolution.

In a practical situation, it may be possible to suggest that the ambiguity may be resolved to some extent by observing the occurrences in the text of the string "Congdon and Lorentz". For example, if the text indicates that this string occurs in several places, then the likelihood of its meaning being (a) above is increased. We may also take a page from the work on information retrieval systems by checking against the

bibliographic references associated with this text. Should Congdon and Lorentz be co-workers, then it is most likely that this will appear under one bibliographic reference.

As far as the English representation of this string is concerned, there is a further cross-check on the number-agreement of the word "experiment". Although this singularity by itself is still ambiguous, it is a possible indication that Congdon and Lorentz together performed this particular experiment. Unfortunately, when no "number word" is explicitly expressed in Chinese, then the noun itself is indeterminate as to its number. The above phrase may well refer to one experiment or several. It seems then that ambiguities of this type are not readily amenable to general rules in the grammar. Specific checks must be built into the system to resolve these semantic problems. At the present stage of research, one can only attempt some "ad hoc" disambiguation procedures, such as those already mentioned. But these first halting steps may become firmer strides as the work progresses.

IV.2.0 Prepositions and Prepositional Phrases

There are two paths being pursued by the Project in dealing with the problem of prepositions. The first one is to incorporate the preposition with certain items and enter it as one entry in the dictionary. This will be the case when the English rendering is idiomatic. The second path is dealing with

Chinese postpositions, such as shang 'on, onto' nei 'within', li 'within, inside', etc. which may or may not be required in the English output. Where it is not required in the English output and where it is possible to find an unambiguous environment in the Chinese structure, we shall implement interlingual rules of deletion directly in the grammar itself. Where such unambiguous environment is not available, then it would be necessary to first reduce the possible alternative prepositions in the dictionary, pick a more 'encompassing' English preposition and then post-edit this result.

IV.2.1 Prepositions in Chinese

In describing the locus of an activity with respect to a particular object, or the locus of existence of such an object, English often makes use of an adverbial phrase formed by a noun (the object) preceded by any of a syntactically unique class of particles called prepositions; e.g. at, in, on, to, for, and so forth. In Chinese no such unique class of particles exists. Instead, prepositional-type relationships are expressed through the use of a loosely-grouped series of

(a) Positional verbs (PV's), which precede the object and generally indicate the motional aspect of an activity with respect to that object, e.g. cong 從 'from', gen 跟 'with' (comitative), yong 用 'with' (instrumental), dau 到 'to, towards', wei 為 'for'

(benefactive). In appropriate contexts, PV's can also be translated into English as verbs; thus, gen 跟, cong 從 'to follow', yong 用 'to use', dau 到 'to arrive, reach', wei 為 'to act as'.

(b) Positional nouns (PN's), which follow the object and generally indicate the stationary aspect of an activity or of a statement of existence, e.g. limian 裡面 'inside' (≠ into), shangmian 上面 'on top of' (≠ onto), houmian 後面 'behind', qianmian 前面 'in front of', waimian 外面 'outside', etc. In appropriate contexts, PN's can also be translated into English as nouns. Thus, limian 裡面 'the insides', shangmian 上面 'the top', houmian 後面 'the back', qianmian 前面 'the front', waimian 外面 'the outside'.

Normally, a prepositional type relationship in Chinese requires one of the following sequences

(a) PV-N-PN

(b) PV-PN

(c) PV-N

An example of (a) would be dau fangzi limian (到房子裡面) (lit. 'to house inside') or 'into the house'. When the relationship is stationary instead of motional the semantically neutral PV zai 在 is used, e.g. zai (PV) fangzi (N) limian (PN) (在房子

裡面) (lit. 'at house inside) or 'inside (≠ into) the house'. As in English, the object can be omitted when understood, which is represented by (b) above: zai (PV) limian (PN) (在裡面) 'inside'.

Finally when the particular stationary aspect of an object is irrelevant (as may be the case with certain motional PV's), the PN may be omitted (the (c) sequence above). Furthermore, certain objects may require, or optionally allow, the absence of a PN even where reference to a stationary aspect is desired.

Thus

PV	N	
zai	Beijing	在北京 'in Peking'
in	Peking	

is acceptable, but <u>not</u>

PV	N	PN	
zai	Beijing	Limian	在北京裡面
in	Peking	inside	

to mean 'in Peking'.

However, both

PV	N	
zai	fanyingqi	在反應器
in	the reactor	

and

PV	N	PN	
zai	fanyingqi	limian	在反應器裡面
in	the reactor	inside	

both mean 'inside the reactor'.

Whereas

PV	N	PN	
zai	fangzi	limian	在房子裡面
in	the room	inside	

means 'inside the room', the sequence

PV	N	
zai	fangzi	在房子
in	the room	

does not.

Let us consider the case where the existence of a definite noun phrase is described relative to a stationary locus. Consider the following examples

(1)	fanyingqi	zai	Beijing	反應器在北京
	the reactor	is in	Peking	
		PV	N	

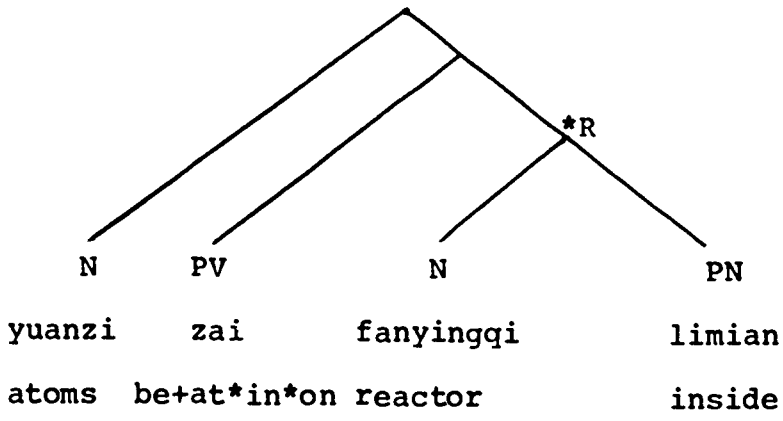
(2) yuanzi zai fanyingqi limian
原子 在 反應器 裡面
the atoms are in the reactor inside
PV N PN
'the atoms are in(side) the reactor'

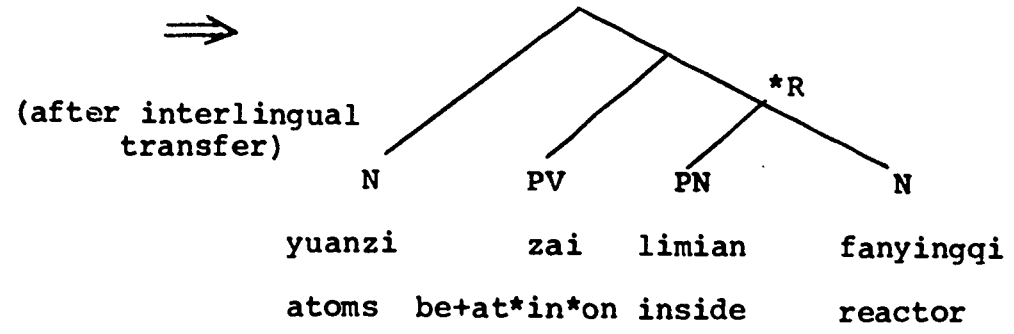
(3) yuanzi zai limian 原子在裡面
the atoms are in inside
PV PN
'the atoms are inside'

Since there is no other main verb in the above sentences, the context is appropriate for selection of zai 'be at, be in, be on' as the PV which supply the verbal element required in the English translation. Our next problem is how to obtain the correct English preposition.

In (1), since no PN is present, the necessary information must be inferred from characteristics of the object noun N itself. Assuming that we had a sufficiently precise categorization of Chinese nouns in general, for example in terms of features, then inspection of the object N would indicate which English preposition type would be needed. (In this case Beijing 'Peking' is itself a locative noun). Insertion of such prepositions could (a) be triggered during or after parsing by the presence of certain specially-marked nodes, or (b) be accomplished by differential glossing of the same character zai 在, in CHIDIC, where each gloss will contain a different

preposition. At present insertion of this type of information is most efficient by direct representation as a CHIDIC gloss since our present grammar-code categorizations of nouns are not yet fine enough to allow unequivocal selection of the required preposition. We have therefore allowed in the gloss itself several alternative choices which could be decided upon at the post-editing stage. For example the above preposition zai will be coded as VG11 and glossed as "be+at*in*on". In sentences (2) and (3), however, where prepositional information is present in the form of the PN limian 'inside', the occurrence of 'at *in*on' in the gloss for VG11 zai would be superfluous and give rise to an undesired interlingual transformation, as shown in the following structure:





giving 'the atoms be+at*in*on inside the reactor' having the undesirable duplication of prepositions. In this case an additional entry for zai, coded VG12 and glossed simply as 'be' is a better solution.

A final point concerns the N + PN combination itself. That is, when appropriately glossed and flipped with the object N, the PN will yield the correct English preposition in its proper place in the English output string. This flip is triggered by an interlingual *R node generated wherever the sequence N + PN appears in the string being parsed. Note also that since PN's must often have different English glosses depending on whether they are preceded by an object N or not, two separate categories, NL11 for the latter case and NL12 for the former, have been set up. Thus, the Chinese PN qianmian will be glossed both as NL12 'in front of' and NL11 'the front'. In the case of limian, the same gloss 'inside' could be used for both NL11 and NL12.

In our interlingual and synthesis work, one of the less developed areas is the proper addition and respelling of such

morphological segments as prepositions. The basic data sets needed are clear. For each morphological addition to be added we need (a) a description of its regular addition to a word and (b) a table of exceptions. In addition an exact formalism will need to be developed to describe how such segments will be combined in the interlingual trees.

The formation of proper English output depends to a large degree on having in the dictionary glosses which can be systematically edited either by machine or by a post-editor. The new structure of disk CHIDIC is being developed to provide just this type of information.

IV.2.2 Deletion of Prepositions

One of the areas in interlingual work where it was found necessary to delete a Chinese lexical item was that of the locative phrases delimited by discontinuous constituents. In particular, the problem of a discontinuous constituent consisting of the sequence: [locative verb + ... + preposition] was dealt with. For example, for

zai ..... shang 在上 'be+at ..... above'
zai ..... nei 在內 'be+at ..... inside'

it is possible to delete the locative verb (glossed as 'be+at') in Chinese and let the preposition carry the burden in the English translation.

Thus

在	木箱	內
Zai	muxiang	nei
be+at	wooden box	inside

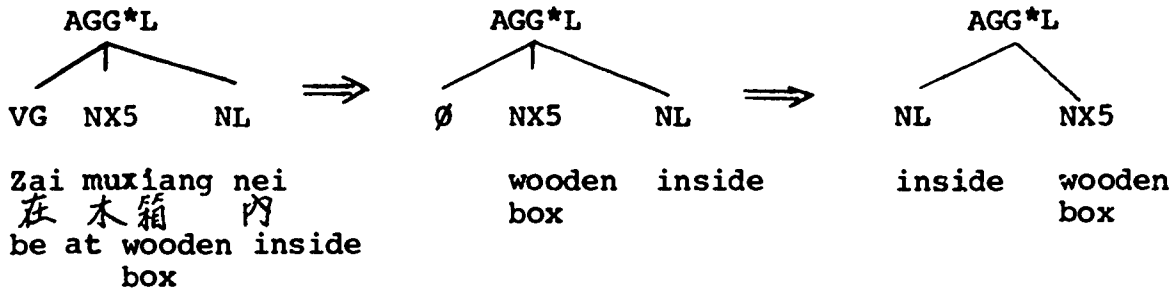
after analysis becomes:

	木箱	內
ø	muxiang	nei
	wooden box	inside

and interlingual processes permute the two constitutes to give us:

內	木箱
nei	muxiang
inside	wooden box

This may be represented by the following structural changes to the string (where AGG*L is the Locative phrase flatted for a left deletion):



This type of deletion of discontinuous elements has been extended to the treatment of certain "absolute phrases" which also exhibit discontinuity. For example, in the following phrase:

凡 大鼠 白細胞 沒 出現 者
fan dashu baixibao mei chuxian zhe

the first and last items: fan .... zhe are discontinuous constitutes but separately are glossed as 'all those' and 'the one that', resulting in the following:

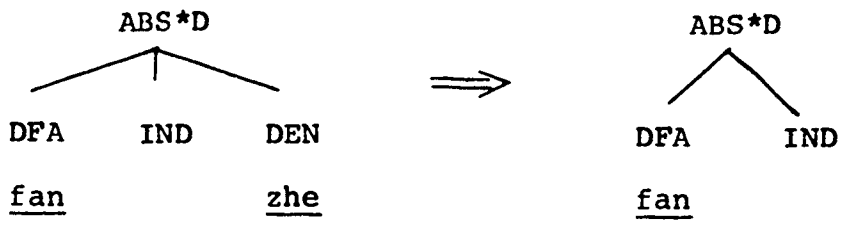
凡 大鼠 白細胞 沒 出現 者
fan dashu baixibao mei chuxian zhe

all those rat white cell have not appear the one that

'in all those (cases where) rat white cells have not appeared'

It is seen that the rightmost item zhe does not directly contribute to the clarification of the English string. Since fan ..... zhe in fact should receive one gloss, it will simplify the English string processing stage if zhe is deleted and a better gloss is given to this discontinuity, e.g. in this case fan can be glossed as 'in all cases where' just in case deletion of zhe occurs.

A rightmost constitute deletion rule (ABS*D) for absolute phrases is as follows:



IV.3 Nominalization with DE

Further revisions of the nominalization rules involving the morpheme de with a view to more direct English output were carried out. Formerly no differentiation was made for the gloss of de, i.e. it has the composite gloss "that * which * of * 's*ø". We have now implemented rules which will automatically choose 'of' when the nouns involved have either the feature, [+Abstract] or [+common] as against human or animate nouns which can take the possessive 's.

e.g.

原素	的	同位素		
yuansu	de	tongweisu		
element	de ↑ of	isotope	-->	isotope of element

愛因斯坦	的	理論		
aiyinsidan	de	lilun		
Einstein	de ↑ 's	theory	-->	Einstein's theory

Furthermore, the relative clause with de is now automatically 'which * that', e.g.

容易	變形	的	物品
rongyi	bianxing	de	wupin
easily	change shape		material

--> material which*that easily change shape

Finally, deletion or zero gloss substitution is implemented for cases where an adjective precedes the noun, e.g.

放射性	的	物品		
fangshexing	de	wupin		
radioactive	de ↑ ø	material	-->	radioactive material

IV.4 Existential Verbs You and Shi

1. Recent analysis of texts which have been submitted for processing indicates an inadequacy in the rules involving the lexical items you 有 'to have' and shi 是 'to be' with grammar codes VY and VC respectively. As in the case with the English 'to have' and 'to be', these verbs when used existentially occur in many different structures. Sometimes they even cross over in their application. For example 'have' and 'there is/are' are rather similar in meaning in English, in sentences such as

(1a) In front is a river

(1b) There is a river in front

and (2a) Next year is the general election

(2b) Next year there is a general election

correspondingly the Chinese sentences with shi 'be' and you

'there is' are as follows.

(1a') qiantou shi yi tiao he 前頭是一條河
front a (classif.) river

(1b') qiantou you yi tiao he 前頭有一條河
front a (classif.) river

(2a') mingnian shi da xuan 明年是大選
next year general election

(2b') mingnian you da xuan 明年有大選
next year general election

In this shi - you alternation, shi can be substituted by you only when two conditions are satisfied

(1) the logical relation between the subject X and subject Y is such that X ≠ Y

and (2) when both shi and you have an existential meaning.

In order to obtain the correct translation for you, the grammar code VY with gloss 'have' is inadequate, since it will render a sentence such as (1b') into

(1c) *In front has a river

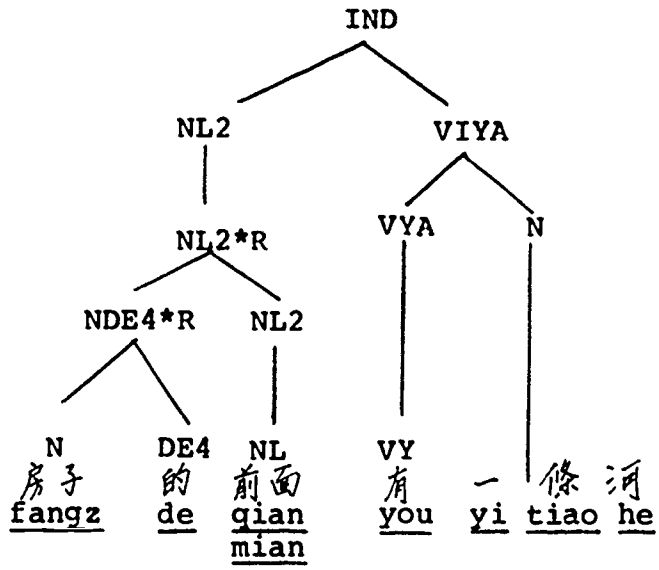
rather than the correct English sentence

(1b) There is a river in front.

or (1b") In front there is a river.

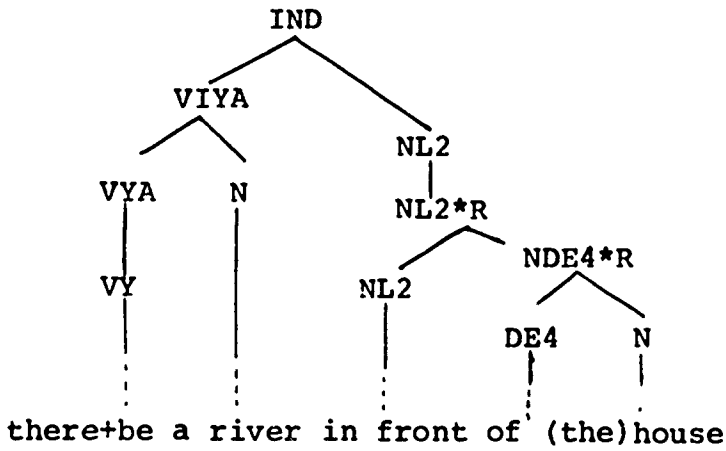
Thus it was necessary to have an additional grammar code VYA for you and glossed 'there+be', which will trigger a series of

interlingual actions that may be represented by the following change in structure:



(the) house of in front there+be a river

which will result in



Rules for the similar case of shi have also been developed.

IV.5 Comparatives

The comparative construction in Chinese is quite regular but differs greatly from the English, thus requiring many English readjustments. E.g. the Chinese sentence

(1)	John	比 bi	Mary	高 gao
	John	Comparative marker 'Compared to"	Mary	tall

must be rendered into English as

John is taller than Mary.

Whereas

(2)	John	比 bi	Mary	聰明 congming
	John	'Compared to'	Mary	intelligent

although structurally the same in Chinese, must be rendered as

John is more intelligent than Mary.

Thus the class of stative verbs (VQ) including gao 'tall' and congming 'intelligent' must be reanalysed in terms of the English output into two separate subcategories. An extensive revision of VQ verbs was carried out and a large set of rules for the comparative was written.

For example, in English we have pairs such as 'TALL' 'TALLER', 'INTELLIGENT' 'MORE INTELLIGENT', 'GOOD' 'BETTER', whereas morphological changes as such do not exist in Chinese. It is therefore necessary to subcategorize VQ into various subtypes in terms of English morphology. The subtypes of VQ suggested

are as follows.

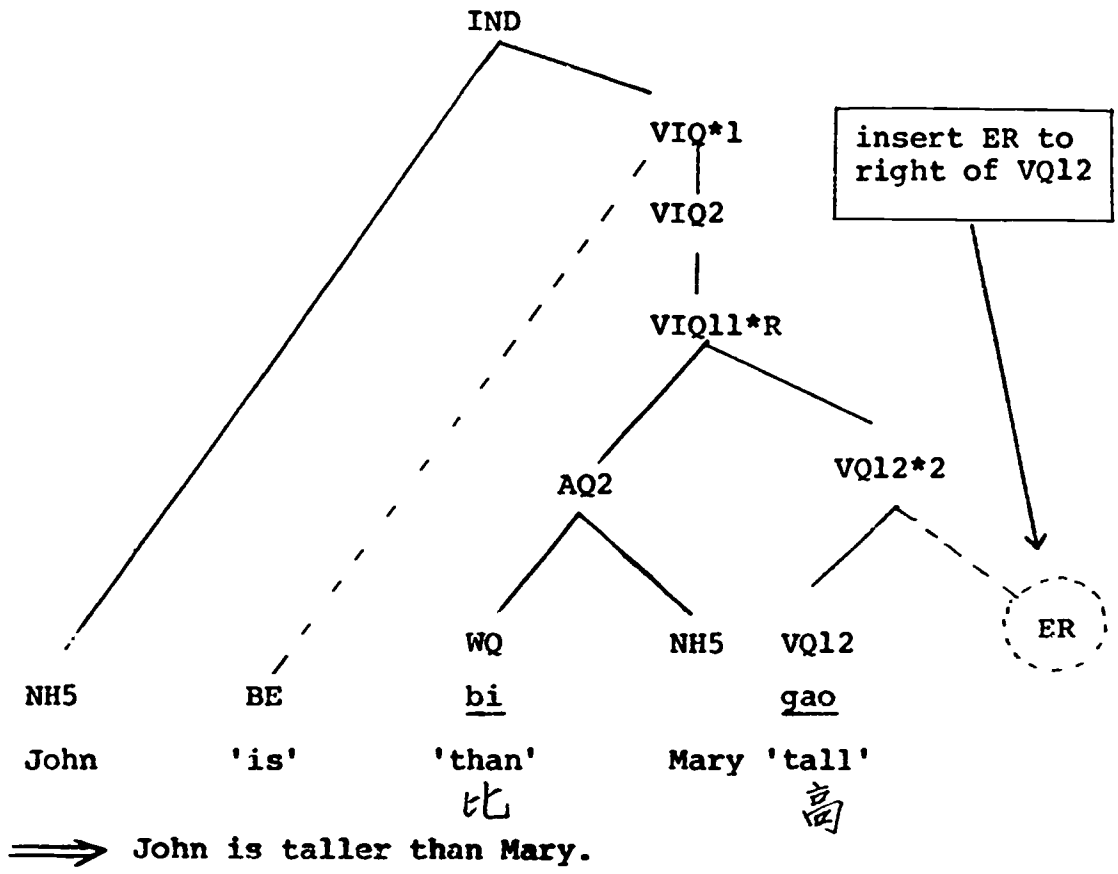
adjectives undergo irregular morphological process to form the comparatives such as 'BETTER', 'WORSE', e.g. VQ11: hao 好 'good', huai 壞 'bad'.

adjectives take ER to form the comparatives such as 'TALLER', 'HAPPIER', e.g. VQ12: gao 高 'tall', yukuai 愉快 'happy'.

adjectives take MORE to form the comparatives such as 'MORE INTELLIGENT', 'MORE ABSTRACT', e.g. VQ13: congming 聰明 'intelligent', chouxiang 抽象 'abstract'.

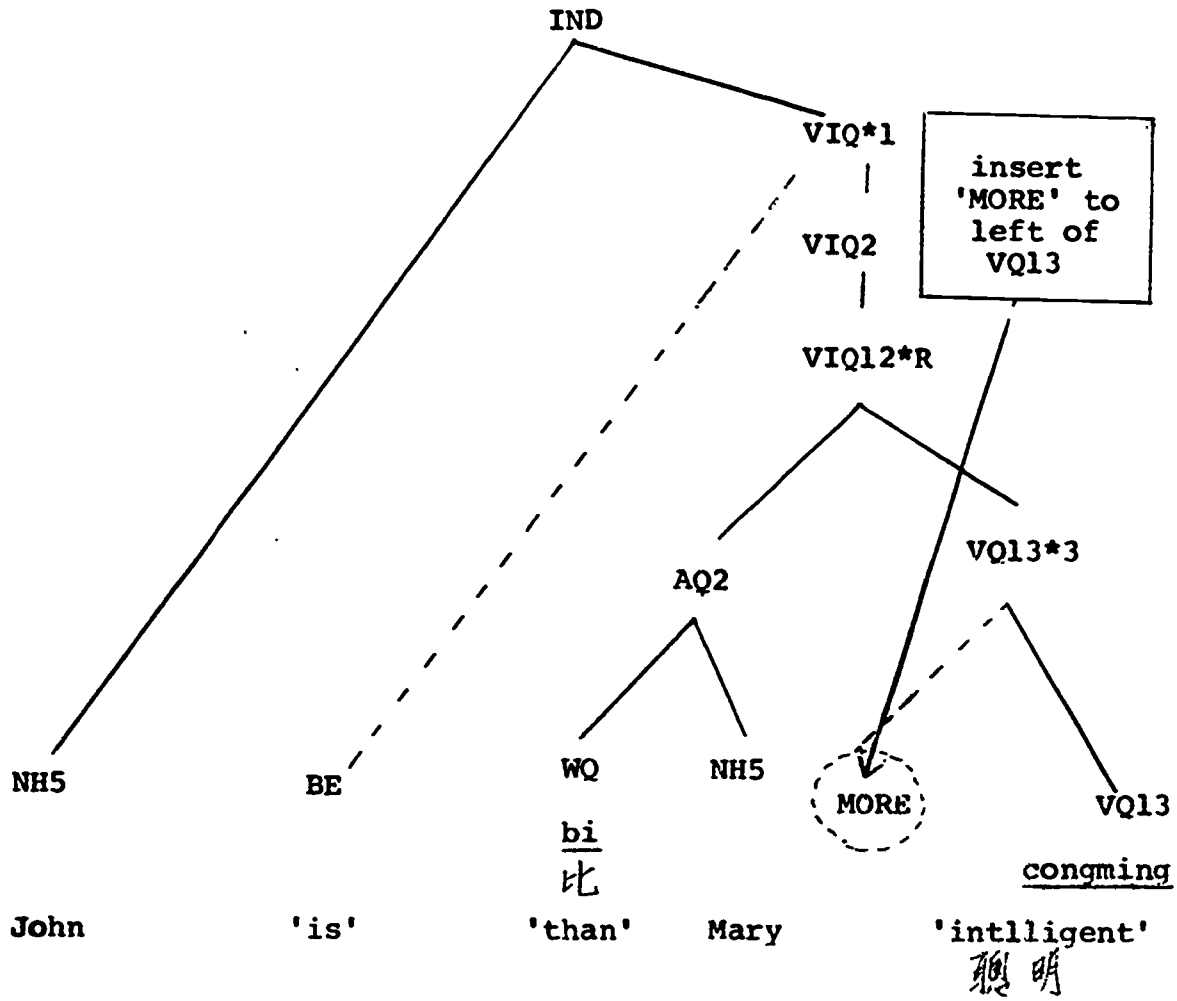
Examples of interlingual rule applications which convert sentences such as (1) and (2) above into the following (1') and (2'):

(1)'



⟹ John is taller than Mary.

(2)'



⟹ John is more intelligent than Mary.

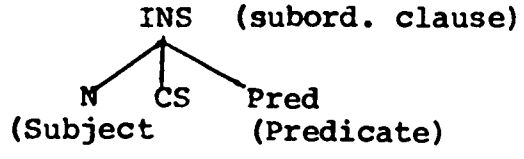
IV.6 Parsing Subordinate Clauses

In general, subordinate clauses have a structure similar to nonsubordinate clauses, except that they are preceded by subordinate conjunctions (indicated by CS in our grammar code). But Chinese subordinate clauses has the extra characteristic that the subject mentioned in the nonsubordinate (independent) clause is very often not repeated again in the subordinate clause; or if the subject is mentioned, the subordinating conjunction may separate the subject from the predicate of the clause. e.g. the sentence

(1) 他 雖然 沒有 告訴 我 可是我已經 知道了
ta sueiran meiyou gausong wo, keshi wo yijing zhidao le
he although have not told me, yet I already know
'although he hasn't told me, I already know.'

where the subordinate conjunction sueiran separates the subject ta 'he' from the rest of the clause - the predicate.

The grammar already has rules which would take care of the juxtaposition of subject and predicate, without the intervening conjunction. There are approximately 50 such independent predicates and, in order to parse just those cases where it is the subject that is separated from the predicate, several hundred rules of the form



would have to be added to the grammar. This is because for each different predicate, separate rules must be written to account for different subjects appearing in the N slot. For example, a predicate like VI3 would require one set of rules:

INS → NN5 + CS + VI3
INS → NN5 + CS + VI3
INS → ND5 + CS + VI3
INS → NF5 + CS + VI3
INS → NH5 + CS + VI3
INS → NR5 + CS + VI3
INS → FN2 + CS + VI3
INS → FNS + CS + VI3
INS → NNS + CS + VI3
INS → NRS + CS + VI3
INS → NDS + CS + VI3
INS → NFS + CS + VI3
INS → NHS + CS + VI3
INS → NBS + CS + VI3

Whereas VIH3 would require a different set:

INS → FNS + CS + VIH3
INS → FN2 + CS + VIH3
INS → NFS + CS + VIH3
INS → NFS + CS + VIH3
INS → NHS + CS + VIH3
INS → NH5 + CS + VIH3

INS → NR + CS + VIH3

INS → NRS + CS + VIH3

Finally the number of rules needed is easily doubled or tripled if we take into account cases of preposed object or object/subject. How does such a state of affairs as the above come about? The answer is simple enough: there is no easy way of indicating the general notion "predicate" in our present grammar. We are instead forced to mention all cases of specific predicates regardless of rule environment.

A solution to the problem may be indicated if we note the restricted environment in which the N-CS- Pred Construction occurs: the N or N's are always immediately preceded by a period or semicolon, and Pred is likely to be followed by a comma. The idea is to institute the following steps during pre-edit: (a) take whatever occurs between the comma and CS up to an "all-inclusive" node which would most closely correspond to Pred itself; (b) Skip over the CS, and (c) take whatever occurs between CS and the period or semi-colon up to another all-inclusive node. Once the string corresponding to Pred has been identified, it will then be possible to check the verb to see what features (+Human, +Animate, +Physical, etc.) it has and determine whether the N preceding CS should be subject or object.

This method would also lend itself well to other cases where multiple combinations of predicates or noun-phrases are possible, for example, rules involving CV, which connect two predicates and rules involving CN, which connect two noun phrases (see section IV.1).

IV.7 Subgrammars and Multiple Grammar Applications

The current form of our grammar has been maintained as a monolithic grammar which will make its best efforts to correctly recognize and parse any text string presented to it as a parse unit. However, within this full set of grammar rules, there are clearly distinct groups of rules which deal with verb complexes, noun complexes, prepositional phrases, adverbial phrases, numerical phrases and so on. Moreover, the formation rules of these complexes also can be distinguished as to their levels of complexity. It has been our experience that, as a result of the parsing algorithm, simplifications in the organization of the full set of rules can result in better parsing results. We can think of the bottom-to-top parsing algorithm as using different sets of rules at different levels to form the tree structure which eventually results in the representation of a parsed string. These different sets of rules then can be considered as "subgrammars" which will apply to a specific type of constituent, such as the verb phrase complex or prepositional phrase. These subgrammars, then, can

be obtained quite directly from the existing 'full' grammar and applied at the appropriate stage. The result is a partial ordering of such subgrammars, whose rules may or may not be ordered internally within each such subgrammar. The order of application of these subgrammars may be specified in advance or may be brought into play on the basis of certain segmentation cues attached to lexical items or to a specific rule.

Let us consider the rules required for parsing numbers in Chinese texts. Actually there are three different sets of rules to be accounted for.

(1) Arabic digits

(2) Chinese digits

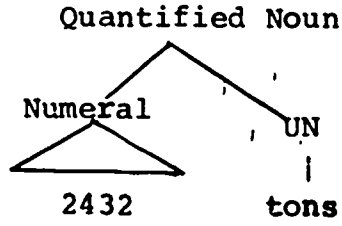
(3) Chinese numeral system

The complexity arises in the second case when Chinese digits are (1) used somewhat like Arabic digits, i.e. as a sequence of digits such as 2432 but written in Chinese as 二四三二 or (2) used in the Chinese numeral system, which has ascending unit quantity sequences equivalent to 1 - 10 - 100 - 1,000 - 10,000 - 100,000, etc. A separate 'small' grammar must be written to account for each of these three co-existing systems. However, once these numbers have been successfully handled, the resulting function of the 'top node' for all of these three numeral types are essentially the same. Thus at the level of the terminals, there will be a choice of 3 subgrammars for application. But at the next level these differences are no

longer relevant and the constituent is just recognized as a simple numeral unit. To illustrate

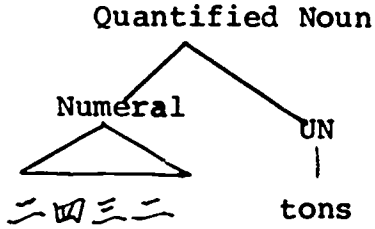
(1) Arabic digits

2432 tons



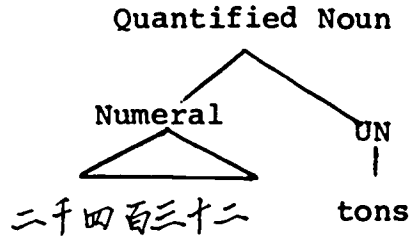
(2) Chinese digits

二四三二 tons



(3) Chinese numeral system

二千四百三十二tons



In each case, the appropriate subgrammar has to apply to the sequence of numerals dominated by the node 'Numeral'. However, the rule at the next level

Quantified Noun → Numeral + UN

is the same for all three cases.

Another case where the separate application of sub-grammars would increase the efficiency of the parsing process is to separate the two styles in written Chinese viz., modern written style and classical style. These can coexist in the same written text, but there are differences in structure and in morphological formation which have to be strictly adhered to even within this coexistence. For example, certain monosyllabic words would be free nouns in the classical style but would have to be bound forms if used in the modern style; otherwise a derived polysyllabic form of the same noun has to be substituted in the same place.

Problems of this sort have been ignored in contemporary syntactic analytic methods, since the concern has been with synchronic grammars. However, this is a very real situation which must be faced squarely by researchers dealing with modern written Chinese texts. It seems to us that the clear separation of these tasks in the grammar rules dealing with separate styles can find a solution by using our concept of subgrammar applications.

During this contractual period, our efforts in this direction were coupled with the segmentation of text strings into smaller parse units, mainly using the comma as segmenter. Higher level rules in the grammar, which should only apply later, were experimentally eliminated from the full grammar and the resultant grammar used as a subset for parsing these units.

As a result, it became much easier to control the rules relating to one subtype of syntactic structure. The discussions on points of analysis have taken this into consideration. It is not possible to obtain directly from the current SAS results of multiple applications of subgrammars, since this would require extensive programming modification to a system whose data base was not originally planned for this purpose. But the partial results obtained on separate runs seems to confirm that this approach is basically sound and an algorithm for multiple grammar applications within the same run is being incorporated into the new system under development. (See also discussion on results of runs of text in this report).

V. ANALYSIS OF TEXTS

Three different texts, totaling about 20 pages (15,000 characters) were subjected to detailed analysis, both as a means of improving the linguistic rules and for vocabulary control of CHIDIC. As a result of changing the segmentation strategy from segmenting on full sentences ending with periods and ignoring all intervening commas to one in which all comma segments were accepted as parse units, there was a decided improvement in parsing success. The three texts, labeled Physics 4, Physics 5 and Physics 6 were run at approximate equal intervals at the beginning, middle and end of the contractual period. Their results are discussed separately below:

V.1 Physics 4

The text identified as "Physics Text 4" was run under two significantly different modes, but using exactly the same grammar version, viz. Version M. The two runs were made to help us identify the ability of the grammar with regard to its handling of long sentences versus shorter sentences. The runs also incorporated the new routines that have been added to the SAS (some of which are discussed in the Programming Section of this report.)

The Project has for sometime been confronted with the problem of parsing sentences of a highly complex nature -

sentences that may very well be considered whole paragraphs. Such sentences naturally tax the ability of any grammar which expects to handle sentences of reasonable length, say something in the order of 20 to 40 Chinese characters long. But because of the presence of many longer sentences, the performance of the grammar deteriorates; and it becomes more difficult to pin-point areas where the optimal improvement could be made. In order to evaluate the performance of the grammar more accurately, it was decided that these overlong sentences ought to be treated in a principled manner as sequences of well-formed shorter clauses. To this effect, our approach was to base text segmentation not only on the periods which indicate end of sentence, but also to segment on commas. Linguistic intuition indicates that a true constituent would not span a string which includes a comma.

With this in mind it was encouraging to compare the results of our two runs. The first run was for the complete text and the second run was only for the first fifth of the text. For the first run there were 87 parse units, obtained by segmentation on <u>periods</u> only. For the second run, the first 17 parse units of the first run were further segmented on <u>commas</u> and <u>periods</u>, giving a total of 51 parse units.

FIRST RUN

# of Parse Units (Segmented on periods only)	:	87
Max. Length of Parse Units	:	152 telecodes
Units Parsed to Nounphrase and/or Sentence*	:	17

* Sentence includes simple sentences or clauses and complex sentences.

SECOND RUN

# of Parse Units (equivalent to parse units 1 to 17 of First Run, segmented on commas and periods)	:	51
Max. Length of Parse Units	:	34 telecodes
Units Parsed to Nounphrase and/or Clause or Sentence**	:	15

** Sentences would normally be simple sentences, equivalent to a clause.

It is clear from a comparison of the two runs that the grammar showed much better performance in the second run, which

is only one-fifth of the length of the first run.

Our decision to run text in the second segmentation mode will provide advantages such as:

(1) more accurate and tighter control of the grammar rules as a whole.

(2) ease in pinpointing weak areas in the grammar.

(3) clearer insight into interlingual processes by concentrating on shorter sentences.

(4) closer approximation to "normal" English sentence length in the translated output.

With regard to the latter two points, inspection of these shorter parse units confirmed that such Chinese clauses are very prone to omission or non-repetition of sentence subject. For example, many clauses will begin with auxiliary verbs or modals such as must, should, possible, etc., where English would require a dummy subject "It" or "It is" to precede the auxiliary. Isolating the Chinese clauses now makes the task of supplying such dummy English subjects, which is well-motivated in any case, a more transparent problem than has hitherto been possible.

V.2 Physics 5

This was the first complete text run under "comma segmentation" mode. There were 384 total segments or parse

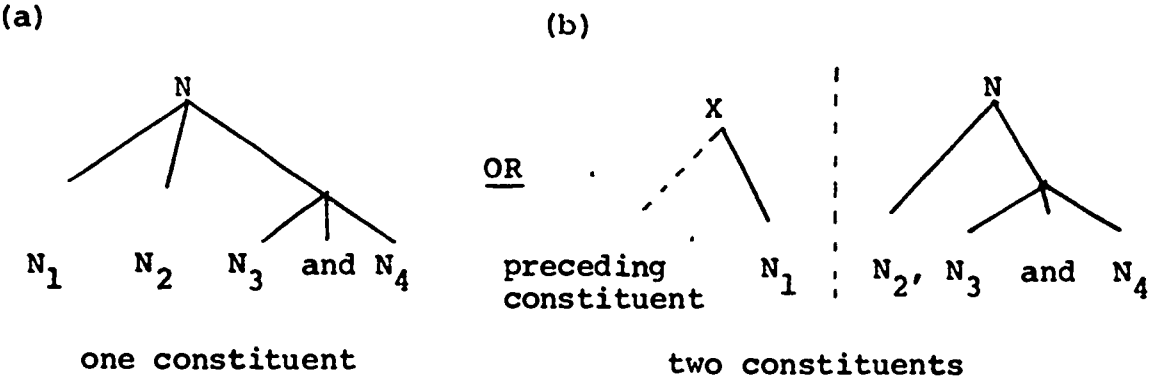
units made in 3 separate physical runs. The gross percentage of units parsed was 75%. Even under comma segmentation, some units were 50 to 60 characters in length. But these were the minority. The majority did not exceed 30 characters in length. Although a conclusion is certainly premature, the statistics on segmentation units (as opposed to sentence length, where each "sentence" is defined as a string ending with a period, and where all commas, etc. are ignored) do seem to suggest that efforts in dealing with parse units of up to 30 characters in length is a practical intermediate cut-off point. Maximal efforts should be concentrated in this area in obtaining a good grammar which can parse a very high percentage of such units. Since other textual factors are involved which the present form of the grammar cannot properly handle, the present form of the grammar must be buttressed with other devises to increase parsing success. Among these would be a systematic use of intra-sentence information in order to enable the system to process more complex sentences. For example, there may be a series of nouns some of which may have been separated by commas while others are not:

(1) N_1, N_2, N_3 ,'he/and', N_4

(2) N_1 ,'he/and', N_2, N_3, N_4 ,'he/and', N_5

Because comma usage is not consistent, it is difficult to decide whether (1) and (2) can each be considered as a single

compound noun phrase, or, e.g. that N_1 in (1) may belong to a preceding constituent whereas N_2, N_3 he N_4 belong to the following constituent, i.e. one has a choice of either



one constituent

two constituents

This type of problem can only be solved after extensive studies of the nature of compounding. The discourse context will also greatly affect the results. However, this is one very real problem that must be tackled in order to decrease the ambiguity problem in general. We feel that lexical heuristics and feature parsing coupled with fuller analysis of CHIDIC entries will be such a first step in the right direction.

Again, it should be pointed out that at this stage of research, a parsed unit is not necessarily equivalent to a correct or unambiguous analysis. Also a parsed segment, when joined to another parsed segment also may not necessarily result in a correctly parsed larger unit unless the intra-sentence information is already available to filter out the aberrant ones.

Unfortunately, the existing SAS is much too rigid to accomodate this type of information. In order to achieve better synthesis of component parse units, the synthesizer portion in the new Parser under development will take this into account. This will be an additional method of solving the complex-compound sentence problem.

Thus although the Physics 5 text gave a general average of 75% of all units parsed, the above observations must be taken into account to balance the picture. The fluctuation in style and content of each text also affects this percentage, as had been discussed in the final report of our preceding contractual effort.

V.3 Physics 6

This text is essentially uniform with the content of Physics 5. However, the parse units in Physics 6 were generally slightly longer in length and the sentence structures more complex. Physics 5 had more short phrases such as subheadings, lists, etc. which were less than 20 characters long.

	Total Sentences	Total Parse Units	Average Parse Units Per Sentence
Physics 5	115	384	3.3
Physics 6	88	423	4.8

Gross parsing percentage dropped to 64%. However, this is not

at all an accurate picture since our run statistics left out many other factors which could not be easily gathered during machine processing. As was mentioned in Chapter III, one miscoded entry in the dictionary would, under this version of the SAS, result in a count of no parse. Since extensive dictionary coverage is a continual effort, the variation due to this type of error should decrease, and is well understood as a problem which is capable of solution. Another aspect of this run was that the emphasis had been to decrease the number of top nodes recognized for each parse unit. The decrease in the number of top nodes for each parse unit is a good indication that parsing has been accomplished with fewer ambiguities than previously. In both texts, units that have been successfully parsed to only 1 or 2 top nodes comprise 70% of the parsed segments. Again this figure must be understood in the light of correct and incorrect nodes. Careful inspection of each parsed segment also indicated greater acceptability of these top nodes as correctly analysed ones. It is in this area that the current efforts in decreasing ambiguity has been showing substantive results.

V.4 Conclusions

One of the major difficulties encountered in these texts is, as already discussed in Chapter IV, the problems of how to handle correctly noun compounding, noun modification and noun conjunction. These already thorny linguistic problems are

further complicated by the fact that many nouns have verbal counterparts - again undifferentiable by themselves because of the lack of morphological markings. Thus in English, where one can speak of an infinitive, a gerund or a participle, in Chinese one should properly only speak of a basically verbal category which, under the appropriate syntactic conditions, would be equivalent in function to one of the three categories in English. From the interlingual viewpoint, these categories in English can be considered as derived from the same basic verbal category, which is the only one available in Chinese. This is an illuminating result for comparative syntax in MT. The unity and simplicity of the Chinese structure splits into several surface forms in English. To force this tripartite structure onto Chinese itself would add unnecessary complications to the analysis of Chinese. However, by a careful understanding of these processes as dealing with Chinese on the one hand and with English on the other, reflecting our 'Analysis' and 'Synthesis' approaches to MT linguistic research, these problems are seen in clear perspective and capable of principled solutions.

VI. PROGRAMMING

Considerations pertaining to conversion to IBM System 360 and English string output capability has led to improvements and redesigns within the SAS presently running on the CDC 6400. Programs are now written with this conversion in mind in order to achieve maximum compatibility by utilizing the minimum of necessary machine dependent programming in the existing SAS.

VI.1 New Routines for SAS

The main Routines which have been added are: Segment, String Extraction, Direct Character Plotting, and Subdictionary Selection.

VI.1.1

SEGMENT, is the set of routines which will eventually replace the existing PRE-EDIT and LOOK-UP routines. It adds flexibility to the system by being able to segment input texts on specified codes (such as any type of punctuation marks) and passes a much better defined string to the Parser. A first version is incorporated into the SAS. (See Chapter 2 for description of its function)

During the evolution of the SAS and its core of primary programs, attention was drawn to the design restrictions imposed

upon it by some of its older prototype-like routines which characterized the initial system. These routines could not have reflected any of the systematic or global design considerations which were later to appear as a result of direct experience and experimentation. In particular, the input interface, known previously as PREEDIT, had none of the flexibility required by the new parser now under development. This routine requires redesign such that it could be extensible within the framework of the new parser and its subsequently new systems architecture.

SEGMENT is designed as a generalized left to right string scanner which will generate segments of the external telecode text as heuristically likely candidates for parsing, using all of the encoded information within the punctuation marks. SEGMENT will also edit the input string of non-essential supra-segmental punctuation, such as parenthesis, and construct a sentence stri-ped of its literal punctuation information associated with a list of spans representing this necessary partition of the sentence up to n-levels of sub-categorization. Hence this lexical scanner represents a "punctual disambiguation process". The scanner defines the unit of the SAS processing cycle by attaching a static sentence number to each such sub-string and further defines the parser subcycle by attaching parse-unit segmentation level information in the span list.

VI.1.2 String Extraction

String Extraction is the process which will extract the English output from the analysed trees after interlingual processes have applied. The routines will present the output in a linear format which can be post-edited. It is one of the final components of the basic Syntax Analysis System, and completes the skeleton of a syntax-based experimental machine-translation program. Its place in the SAS is as follows: up till this time, the output of the SAS has been a set of trees representing parses and transforms of parses which occurred under a most-highly-valued top node. Many of these parse trees differ from one another in their structure (and thus are necessary for the continuing improvement of the grammar) but do not differ in the strings of terminals which they comprehend. For machine translation output, the crucial information is the different sets of terminal nodes in the trees. Thus, string extraction is the process of deriving the distinct sets of terminal strings from the sets of SAS trees developed during parsing, using but eventually discarding the structural information.

The string extraction component of the SAS (STREXTR) is by far the largest single logical phase of the SAS. It currently contains over 6,000 Fortran source cards, as compared to about 4,600 for all the rest of the SAS including all the plotting, the Graphic Display System source routines, and the

utility library programs.

STREXTR has been designed and coded in a highly modular organization; it currently contains 130 subroutines. The great bulk of the code is machine-independent, with all the 6400 dependencies and formats grouped into a few places for easy changes. STREXTR uses no fixed storage locations, but instead organizes all its data into a collection of stacks, trees, and general list structures. STREXTR does no sorting or searching of sorted tables, but instead does all its table look-ups by address calculation ("hashed" storage)--sometimes straight indirect address calculation, sometimes doubly-indirect, treating the calculated addresses as list heads. The result of this organization is that STREXTR is conceptually very clear, and very easy to modify and change as the Syntax Analysis System evolves.

The need for string extraction arises in the first place because of the interlingual operations carried out after a sentence has been parsed. Since each node in a Chinese parse tree contains information relating it to all other nodes, collapsing a Chinese parse tree into its string of associated terminals would be a relatively straight-forward and well-understood process. But the interlingual operations change this structure in generally unpredictable ways, so the structure of the associated string has to be recovered from the tree anew. The situation is complicated further by the observation that

there is often not a single correct result, but rather a set of possible results which are logically equivalent--they differ, however, in being more or less compact and more or less easy to read. For this reason a large number of decisions about how to proceed have to be made dynamically, and can only be based on heuristics.

A general working approach to the problem would be to fully expand all alternate trees for a structure and then re-collapse them. Unfortunately, this would so explode the size of the intermediate results that it is computationally wholly unfeasible. Thus, STREXTR proceeds by undoing a bit of the logical abridgement of the trees, looking for situations where common subtrees can be seen and collapsing them, and then returning to undo a bit more of the abridgement and repeating the process until new opportunities cease to arise. In this way maximum advantage is derived from the notation developed during parsing, and duplications are always located at the most insightful point.

STREXTR begins by going through the sequential tables created by uprooting a set of trees which have a common top node, and forming them into a linked-list representation. During this process all nodes which can be shown not to have the potential to influence the structure of the extracted string are deleted. General trees are stored, in terms of their "equivalent binary trees." In addition to the regular set of

links to sons and brothers, the abridging pointers are changed so as to provide a direct pointer to the expansion of each node in the tree.

Once this is done, the process of extraction begins in earnest. There are two processes involved, which can be overlapped in execution but which are logically separate. The first consists of finding labels of subtrees which immediately dominate only labels and references to labels, and replacing the father by his sons. This must be done for each reference to the label/father in the tree; taken altogether, this is the gradual unwinding of the list structure.

The other process consists of looking for one of three situations: (1) Two of the n alternative developments of a node are identical. (2) All n of the n alternative developments of a node have a common initial or final sub-part. (3) Two of the n developments of a node have a common initial or final sub-part. The courses to be taken in each of these three situations are: (1) delete the whole repeated subtree; (2) "lift" the common parts of all alternatives up out of the alternative, adjoining them (left or right) to the node which summarizes the alternatives; (3) permute the partially-identical alternatives so as to make them adjacent, create an additional sub-alternative structure over them, and then lift the common parts from them as in (2). It should be clear that these actions have been described in the order of decreasing pleasantness: the

first one gets rid of a lot of structure very cheaply, while the last one creates more structure and is extremely hard to do (though it may open up better opportunities, which is the reason for doing it). Naturally, the attempt is made to use cheap operations before expensive ones are invoked.

After from one to about fifteen passes over a tree-set, all the possible extraction has been completed. The remaining task is to format an output string and print it. It is not quite accurate to describe the output as a "string"; it will still contain, in general, alternatives embedded within alternatives. But since the labels on the remaining tree nodes have no importance, the result can be given a linear representation as a parenthesized string of English words. It would be possible to expand this into the set of strings which it represents, but the result would be less insightful than the parenthesized version, which minimizes the domain of different readings, and which shows their mutual dependencies clearly. At this point the English glosses are retrieved from the dictionary by reference to the dictionary addresses carried in terminal nodes.

But even when this has been done, there may still remain alternatives for the wording of the English. Some of these alternatives represent real ambiguities in the Chinese sentence, which a human translator might or might not be able to resolve by using his general knowledge of the world and of the text

being translated. These real possibilities for multiple meaning in the Chinese should be preserved by the system. Other alternatives do not represent Chinese alternatives, but simply reflect inadequacies in the lexicon or the grammars which have failed to make enough distinctions to permit the system to resolve all the choices. For example, some English noun phrases should use the word "atom"--atom bomb, atom smasher, and so forth--and others should use the adjective form "atomic"--atomic mass, atomic fission, and the like. The distinction between these is one which the system is not always able to resolve, and where it cannot do so it must preserve the alternative forms for the output, such as that represented by the structure in Figure 5.

Once the process of extracting the English words is complete, the final output editing of the English string must be performed. The various distinctions of number, person, tense, etc. which have been gathered must be used to "spell out" the form of each word correctly, adding 's' to plural nouns, adding '-ed' to past verbs, and other more complicated details of the way English words require these features to be shown. This process is impossible to carry out properly for all words, or at least it seems to be so given the current state of our knowledge; but what has been gathered can be used, after which the completed sentence or sentences can be added to the text of the translation being produced.

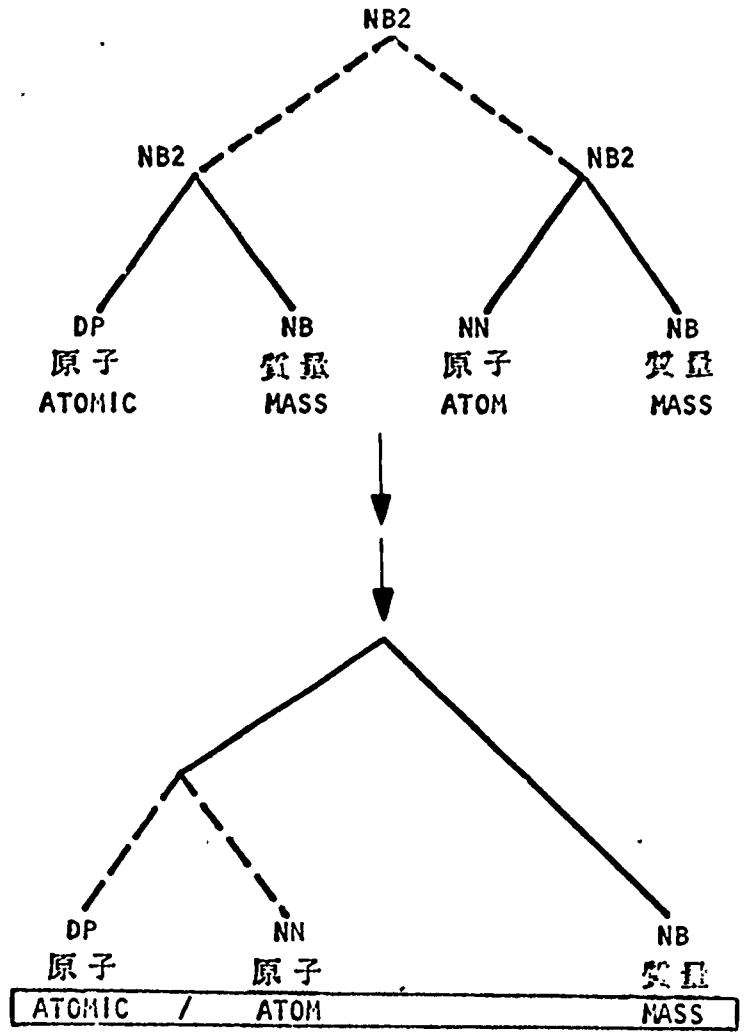


Figure 5. Collapsing of Ambiguous Sub-Trees

The string extraction program is now run as a separate and last component of the SAS. It now extracts English strings and prints them out in a parenthetic notation. Because of the many alternative outputs that may be possible for a Chinese sentence, we still have to make further revisions in this program in order to facilitate the post-editing task. A full string expansion as opposed to the parenthesized format is also being considered. However, if there are a large number of alternative expansions which differ perhaps in only one or two words, this may not be a very economical output format. We expect to make further refinements in this program during the next contract and also attempt full string expansion after the algorithms for extraction have been fully checked out.

During this period the String Extraction segment of the system has shown most improvement. One reason is that this is a new component already written with our IBM/360 conversion compatibilities in mind. It has avoided many of the restrictions of the earlier SAS and is thus capable of continual improvement.

VI.1.3 The Character System

Character System is a set of routines which will store characters to be plotted in the Extended Core Storage of the CDC 6400 and be ready to be used for plotting on peripheral plotters such as the Stromberg Carlson 4020 microfilm plotter,

the Calcomp plotter or other microform systems. This will be especially efficient in plotting characters for concordances. These routines could not yet be completed at the time of this report since system support at the Computer Center have not yet been completed. The routines adapt the basic Kuno character vector sets for more flexible plotting on our system. After adaptation each character is represented by a string of end points packed six to a word in ECS. The address and length of this string are put into a head word which is accessed by the telegraphic code of the character. A block of 9999 consecutive words of ECS is set aside to contain the head words for the 4 digit numeric telecodes. In this case the telecode itself is the index into the table of heads. For the few telecodes which do not consist of four digits the heads are kept in a hash table. In addition to the ability to add, delete or replace entries, a provision is made to reassign a representation to a new telecode.

CDRIVER is the main program of the character system. It calls CONFIG to allocate drives and ECS space and initialize the hash table. CDRIVER then reads a lead card of eight parameters. Each parameter is tested by CDRIVER which then makes the appropriate subroutine calls. The lead card parameters are as follows:

If LDCD (1) equals zero an adapted character dictionary is read, else an adapted character dictionary is created from a

Kuno tape.

If LDCD (2) is not zero, telecode substitution is done.

If LDCD (3) is not zero, update cards are adapted.

If LDCD (4) is not zero, an output dictionary tape is written.

If LDCD (5) is not zero, an output dictionary is printed.

If LDCD (6) is not zero, lines of text telecodes are read and their characters are looked up.

If LDCD (7) is not zero, the vectors for the text are printed.

If LDCD (8) is not zero, the vectors for the text are written onto tape.

VI.1.4 Subdictionary Selection

The current Syntax Analysis System requires that the dictionaries it uses be of a restricted size. The subdictionary selection process consists of taking a text and a large dictionary, for instance our CHIDIC, and selecting from the dictionary all entries relevant to the text. The important considerations are (1) getting all relevant items from the dictionary and (2) insuring that the number of entries selected does not exceed the capacity of the Syntax Analysis System.

The subdictionary selection package has been rewritten to minimize the number of extraneous entries selected, while

maintaining the speed of execution of the old package. A program to select only the relevant items could be written, but it would execute far more slowly and the payoff would be small.

Subdictionary selection consists of three jobsteps submitted together as one job. Jobstep one is the GOALS program and its associated subroutines. This routine scans the Chinese text, performs telecode substitutions, and writes records consisting of consecutive telecodes from the text on a temporary file. These records are called search goals. Jobstep two is sorting these search goals using the CDC 6400 sort/merge utility. Jobstep three consists of the SELECT program and its subroutines; these routines scan the sorted list of telecode pairs and an input dictionary in tandem and write on an output file those dictionary records whose telecode field matches a search goal. Dictionary records with one telecode in the telecode field must match the first telecode in some search goal. Two telecode dictionary entries must match the first two telecodes of a search goal. Dictionary entries of 3 or more telecodes must fully match a search goal, i.e. the first 2½ telecodes or 10 characters in the telecode field must match.

VI.2 Revision of the Parser

This revision of a major section of our programming effort is the result of our experience with the present system. Our experience with the output produced by the SAS have

suggested several improvements that will facilitate the task of parsing input sentences correctly. But we would not wish to rashly launch into any large-scale revisions that are not within the state of the art nor overtax the available manpower.

One of the major tasks envisioned is to produce a more efficient parser. There are different aspects to this efficiency requirement: (1) output of relevant data and ability to select and use them efficiently by linguists of the Project, and (2) efficiency of the parsing algorithm itself.

Taking up the second point first, a more efficient parsing algorithm will be the first concern in improving the parser. Our grammar, as has been noted so often, is a modified context-free phrase structure grammar. Addition of a feature handling capability will impart to it certain context-sensitive characteristics. More specifically, parallel with our task of feature implementation in the grammar and dictionary, the parser must have the capability of manipulating such features during the parsing stages. This is by no means a trivial task. Successful implementation of the feature handling capability also calls for a gradual conversion of the present format of CHIDIC in order to accommodate the feature matrices.

Finally, we wish to be able to parse a string by a "re-entrant" process. That is the output of an earlier parsing will become the input to a later stage of parsing. In other words,

if we had obtained a tree from a first stage parsing, this same tree will be used again in a second stage parsing in order to give a more refined tree. This "re-entrant" concept is quite akin to that implemented in many multiprogramming systems.

Regarding the first point, it has been found that the vast amount of paper output by the present system not only increases the processing time (and thus the cost), but the kind of diagnostic data accompanying the analysis of each sentence are often not necessary for the linguists who will be going over the results of the parsing.

We have therefore implemented a set of options for the final output. For example, one of the most unwieldy sections of the output is the printing of the constitute tables. These tables are an extremely useful diagnostic for checking ambiguous parsings. However, because of the linguist's familiarity with the grammar itself, it is not always necessary to laboriously check tbrough these constitute tables to arrive at answers. We would therefore want to save all these diagnostics on tape and only request for them at another time when it is found necessary to resolve certain complex problems of analysis.

As an atternative, we have made extensive use of the Break Table display as a diagnostic shortcut. The following is a representation of the information provided by a typical Break Table for a sentence 12 characters in length:

Break Table

Sentence 1:

Segment 1	Segment 2
Sentence Position: 1 to 2	3 to 12
1 Partition into 1 subsegment	2 partitions into 2 subsegments
Subsegment 1	Subsegment 1
Sentence Position: 1 to 2	3 to 3 4 to 12
Constituents AA	AQ VXU
AV	WQ
AGG	
	Subsegment 2
	3 to 6 7 to 12
	VIN3 VXU
	VIQ*R VTH3
	NB5

In this table, the 12 character sentence was found to have a major syntactic break occurring between sentence position 2 and 3. In the first segment the possible constituents that could span positions 1 to 2 are either AA, AV, or AGG (representing different categories of adverbials). In the second segment there are two further subsegment breaks. The first subsegment has a break after position 3. If the analysis of the constituents is correct, then either there is no rule such that

$X_i \rightarrow AQ + VXU$ or $X_j \rightarrow WQ + VXU$

and also

$Y_i \rightarrow AA + \begin{Bmatrix} X_i \\ X_j \end{Bmatrix}$ or, $Y_j \rightarrow AV + \begin{Bmatrix} X_i \\ X_j \end{Bmatrix}$, and so on.

Subsegment 2 indicates that there is another alternative analysis for Segment 2 and again indicating the possible constituents which could be obtained. Thus it is possible by inspecting the table, not only to add or delete rules in the grammar, but also to get a good idea of the ambiguous structures which this sentence gives rise to. The table also highlights certain structures which are problematic. For example, in Subsegment 2 the constituents which span positions 7 to 12 indicate this substring could be a verb phrase (VXU or VTH3) as well as a noun phrase (NB5). Intuitively this is rather unlikely, so that the linguist must reconsider the existing analyses for this construction. It is also possible that an incorrect assignment of a grammar code to a particular entry in the dictionary was the problem. Additionally, the lexicographer might discover that this particular string requires the assignment of a grammar code which was previously overlooked. In general, it is the case that there are few trivial problems connected with the break tables. Each break requires careful reanalysis by the linguist.

VI.3 Towards Conversion to IBM System 360

VI.3.1 Machine Independence

During the period under report, special attention was focused on the methods of effecting a smooth transition in the conversion of the current system run on the CDC 6400 to one running on IBM 360/65. The conversion task in this period was characterized by design considerations in the new system for compatibility with 360 conversion. It is highly desirable that the research and converted systems march in step so that results of research can be incorporated into the initial capability system. Conversion then does not mean merely taking the existing working system on the CDC 6400 and converting it to IBM 360 since the former is under continual development. The task of conversion will be simplified if as much machine independence as possible is required in implementing the programs without seriously affecting the efficiency on either system. To this end, one of the basic requirements is that coding in FORTRAN be restricted to a subset language which is as close as possible to standard ANSI Fortran. For a large system that is already in operation, optimization considerations make it impractical to code the complete system in Fortran. Thus assembler language routines are necessary, though these will be at a minimum. Our approach to standardization will be in terms of 3 types of coding:

Type 1 Standard ANSI Fortran.

Type 2 Fortran incorporating extensions of each computer system.

Type 3 Assembler language routines, which are different for each machine.

Type 2 programs present the greatest difficulty in conversion since they have built-in incompatibilities for each system. Therefore extra care will be taken to minimize the writing of these type 2 programs. Type 1 programs should be executable on any ANSI Fortran compiler. Type 3 programs must be written separately for each system. However, since each is written independently of the other system, the conversion problems per se are not as complex as those of Type 2 programs. It is here that the problems of proper interface between modules must be tackled.

Our aim is to program as much as possible in Type 1, supplemented by Type 3 and least in Type 2. Section VI.3.3 is a more detailed description of our restricted subset of ANSI Fortran.

VI.3.2 Structural Programming

Another important aspect of programming design for the new system and its 360 conversion deals with the state of the art concepts on what has come to be known as structural programming as exemplified in the recent works of Dijkstra

(1969), Wirth (1971), Knuth and others. Roughly speaking, structured programming is a discipline intended to support the production of correct, understandable programs which are easy to modify and maintain. It involves the decomposition of a program into manageable units called modules or segments. The program is constructed in an orderly way: The first code written is the very "top" of the system or program; it describes the relationship among the major functional components of the program. The code constitutes a structured program module. The components are represented in the code by writing their module names. The module can be viewed as a program written for an abstract machine. However, since a machine with such high level instructions is unlikely to exist, the next step is to select a module name and code the module which explains it in terms of other module names. This process of going 'downwards' continues until each module name not supported by a module corresponds to an instruction on an abstract machine which exists by virtue of hardware, or supplemented by software. Typically, such program modules are not long and complex, so as to make modification simple. The input to a module and output from it are unique. No goto statements are ever used in order to preserve this unique in and out property. We see this programming discipline as something which has close similarities to the way linguistic analysis of sentences are carried out (cf. the discussion on subgrammars) and is thus a highly valuable method of implementing a machine translation

system. Programs which are presently being written for the new system will adhere as far as possible to this methodology. In order to enable consistent code in our restricted Fortran to be produced in this way, a high level preprocessor (called GASP) has been written to process the code into Fortran.

GASP incorporates an extension to the Syntax of Fortran and has been made to provide for a variety of structured control statements such as If... Then... Else, While... Do, Case... Of, and a number of others. Apart from I/O, the statements of the extension replace all Fortran statements except for declarations, assignments, and subroutine calls. In particular, no go-to statement is provided. This pre-compiler translates the control structures into an ANSI Standard Fortran subset (see appendix to this chapter). GASP is one of several inter-related programs to facilitate the rapid production of large machine-independent software systems for research in MT, and so incorporates the machine-independent manipulation of structured data. This has proved to be highly successful in practise, saving many valuable hours of programmer time which would otherwise have to be spent in laboriously hand coding directly into Fortran.

Our schedule for conversion is to rewrite the system under the new design so that it will execute and output at a minimum the same results as the current system. In the reprogramming of this new system there will be "hooks" where we

can hang on further modules which we plan to incorporate at a later stage but which would delay the conversion task were they to be incorporated during this contractual period. For example, the initial converted packaged will have the capability of accepting input from a dictionary that will contain lexical disambiguation procedures and feature checking mechanisms once these latter tasks have been implemented and sufficiently well tested. However, the work involved in implementing lexical disambiguation procedures and feature checking procedures must result from detailed linguistic study, coding and testing. This again will further delay the total conversion task. Thus we look upon these additions to the system as new system routines which will be implemented gradually in further research efforts.

VI.3.3 Classification of Program Types for the New System

Presented below are descriptions of schemata for various program types, listing exhaustively the Fortran statements which are permitted in each. This listing by statement-type is simply for reference, and does not adequately capture the real point: Type 1 programs should execute properly on any ANSI Fortran Compiler for any machine (having at least 32-bit words, at most 8-bit characters) with no textual changes whatever. Type 2 programs are permitted to give different results on different hardware. Type 2 programs should be limited because each and every one of them they have to be rewritten, and so the rule is: do every possible function in Type 1 programs, calling

on extremely brief Type 2 programs only to isolate machine dependencies.

The following points are to be noted:

1. Although Type-2 routines can be machine dependent, they should avoid exploration into the nooks and crannies of each compiler (in particular, the CDC 6400 RUN compiler). At the moment, the only ways in which Type 2 routines are freer than Type 1 routines is that they may contain

(a) octal constants (only in DATA statements)
(b) masking operators
(c) calls to the shift routines (forbidden in Type 1)
(d) Data statements with implied do loops
(e) type real variables

In many ways, the aim should be to have only Type 1 and Type 3 routines.

2. Not every Type 2 routine should do I/O, and only special Input/Output Type 2 routines should do it--and they should do nothing else.

3. No constant (integer, hollerith, real, octal) should ever appear in any executable statement--as a matter of fact, constants should only be used in Data, Dimension, Common, Equivalence, and Integer declarations. There are precisely two

exceptions to this remark. (a) the integer constants zero and one may be used anywhere, if necessary; (b) error numbers are considered purely local housekeeping, and may appear in statements such as ERRNUM = 4. (b) does not affect the following remark--nor does (a).

4. Actual arguments of procedure invocations may never contain constants, nor expressions involving constants.

5. All non-executable statements must precede all executable statements, so as to give *COMDECK ENTRY control over the insertion of both kinds of statements into the standard prologue. (Exception: FORMAT() statements in Type 2 Input/Output routines.)

6. Subroutines may alter the values of their parameters, or of non-local variables. Functions may alter only the values of variables local to themselves, never the values of their parameters.

7. Do not use type Logical variables, nor logical constants. This means that all tests will necessarily involve the logical relational operators .EQ., .NE., .GT., .GE., .LT., .LE. Logical tests are done with .EQ. and .NE. against the integer 1 (which always means yes, true, etc.) and the integer zero (which always means no, false, etc.) Test functions accordingly return 1 for true or yes, 0 for false or no.

8. The only valid combinations of operands for the relational operators, in our subset of ANSI, are Integer-Integer and Real-Real. (Naturally the occurrence of Reals is highly restricted.)

9. Do loops should always be preceded by one of two things: either a test to be sure that the loop should be executed at all, or else a comment explaining why the loop is tested at the bottom.

10. Never use "extended range" in Do's.

11. Never alter the values of any of the Do limits or of the index variable within the Do loop (which is non-ANSI).

12. Never use a single statement number to terminate more than one do loop.

13. Never do mixed-mode assignments (which are ANSI in a restricted way--contrast mixed-mode operands of arithmetic operators, which are not ANSI at all). The few possible occasions for it arise with reals, for which we use the ANSI intrinsic functions IFIX() and FLOAT().

14. The only valid ANSI Fortran subscripts are (i is an integer variable, c and integer constant): i, c, i+c, i-c, c*i, c*i+c, c*i-c [total forms: 7].

15. Never do statement-number actual parameters, non-standard returns to locations passed in that way, or multiple

entry points. Each routine has a single entry point, the one supplied by *COMDECK ENTRY.

16. The same standard should be adopted for returns, giving each routine a single return. This can be located at the physical end of the texts of routines, and a standard epilogue *COMDECK EXIT can be written.

17. Some miscellaneous things not allowed:

(a) 7-character names

(b) mixed-mode arithmetic

(c) non-standard library functions

(d) namelists

(e) PRINT, READ, PUNCH statements (not FORTRAN IV even)

(f) PAUSE, STOP, etc--except in the unique Main Program

Also, among fortran statements do not use

(g) Statement functions--because they cannot be used with our approach to a standard prologue, since they have to occur just at that executable/non-executable interface which we wish to give only to *COMDECK ENTRY.

(h) Go-to Assignments--because assigned go-to's are not used.

(i) Assigned go-to's--because they are useless.

Instead, use a computer go-to or a SWITCHON CASES statement.

(j) logical assignments--because type logical variables are forbidden.

(k) since there are no logical, double, or complex variables if follows that write logical, double, or complex functions should not be written.

18. Every routine should have calls to two comdecks--one with comments identifying its author, the other with comments identifying its type. With this requirement, we can selectively compile our decks. To facilitate this, TYPE "O" has been named to identify routines which require Gasp processing before they become Type 1.

APPENDIX

Schema for Type 0 Fortran Programs

head line:	Integer function	INTEGER FUNCTION f(a_1,a_2,...,a_n)
	Subroutine(args)	SUBROUTINE s(a_1,a_2,...,a_n)
	Subroutine	SUBROUTINE s
prologue:	Integer declare	INTEGER v_1,v_2,...,v_n
	External	EXTERNAL v_1,v_2,...,v_n
	Dimension	DIMENSION $v_1(i_1)$,...,$v_n(i_n)$
	Common [*COMDECK]	COMMON /x_1/l_1,...,/x_n/ln
	Equivalence	EQUIVALENCE (l_1),...,(l_2)
	Data	DATA $l_1/c_1/$,...,$l_n/c_n/$
*CALL ENTRY		
C GASP		
executable:	Arithmetic assign	v=e
	Logical if	IF(le) S
	Go-to	TO TO k
	Continue	CONTINUE
	Subroutine call	CALL s(a_1,a_2,...,a_n) or CALL s
	If then else	IFTE(le) THEN {S} [ELSE {S}]ENDIF
	Unless then else	UNLESS(le) THEN {S} [ELSE {S}]ENDUL
	While do	WHILE(le) DO {S} ENDW
	Until do	UNTIL(le) DO {S} ENDU
	Repeat dowhile	REPEAT {S} DOWHILE(le) ENDR
	Repeat dountil	REPEAT {S} DOUNTIL(le) ENDR

	For while	FOR v=i [TO j] [BY k] [WHILE(le)] DO {S} ENDF
	For until	FOR v=i [TO j] [BY k] [UNTIL(le)] DO {S} ENDF
	Switchon cases	SWITCHON v INTO $m_1,m_2,\ldots,m_j$ CASEL {S} CASE2 S ... CASEj {S} CASED S] ENDC
epilogue:	Return	RETURN
	End	END

Schema for Type 1 Fortran Programs

head line:	Integer function	INTEGER FUNCTION f(a_1,a_2,...,a_n)
	Subroutine(args)	SUBROUTINE s(a_1,a_2,...,a_n)
	Subroutine	SUBROUTINE s
prologue:	Integer declare	INTEGER v_1,v_2,...,v_n
	External	EXTERNAL v_1,v_2,...,v_n
	Dimension	DIMENSION $v_1(i_1)$,...,$v_n(i_n)$
	Common [*COMDECK]	COMMON /x_1/l_1,...,/x_n/l_n
	Equivalence	EQUIVALENCE (l_1),...,(l_2)
	Data	DATA $l_1/c_1/$,...,$l_n/c_n/$
*CALL ENTRY		
executable:	Arithmetic assign	v=e
	Arithmetic if	IF(e) k_1,k_2,k_3
	Logical if	IF(le) S
	Do loop	DO k i=m_1,m_2[,m_3]
	Go-to	GO TO k
	Continue	CONTINUE
	Subroutine call	CALL s(a_1,a_2,...,a_n) or CALL s
	Computed go-to	GO TO (k_1,k_2,...,k_n),i
epilogue:	Return	RETURN
	End	END

Schema for Type 2 Fortran Programs

head line:	Integer function	INTEGER FUNCTION $f(a_1,a_2,\ldots,a_n)$
	Real function	REAL FUNCTION $f(a_1,a_2,\ldots,a_n)$
	Subroutine(args)	SUBROUTINE $s(a_1,a_2,\ldots,a_n)$
	Subroutine	SUBROUTINE s
prologue:	Integer declare	INTEGER $v_1,v_2,\ldots,v_n$
	Real declare	REAL $v_1,v_2,\ldots,v_n$
	External	EXTERNAL $v_1,v_2,\ldots,v_n$
	Dimension	DIMENSION $v_1(i_1),\ldots,v_n(i_n)$
	Common [*COMDECK]	COMMON $/x_1/l_1,\ldots,/x_n/l_n$
	Equivalence	EQUIVALENCE $(l_1),\ldots,(l_2)$
	Data	DATA $l_1/c_1/,\ldots,l_n/c_n/$
	Data--implied do	DATA $(v(i),i=c_1,c_2[,c_3])/c/\ldots$
*CALL ENTRY		
executable:	Arithmetic assign	v=e
	Arithmetic if	IF(e) k_1,k_2,k_3
	Logical if	IF(le) S
	Do loop	DO k $i=m_1,m_2[,m_3]$
	Go-to	GO TO k
	Continue	CONTINUE
	Subroutine call	CALL $s(a_1,a_2,\ldots,a_n)$ or CALL s
	Computed go-to	GO TO $(k_1,k_2,\ldots,k_n),i$
epilogue:	Return	RETURN
	End	END

Other Type of Programs, in brief.

Type Blockdata-1: head line: BLOCK DATA name

declarations: Integer declare

Dimension

Common [*COMDECK]

Equivalence

Data

epilogue: END

Type Blockdata-2: headline: BLOCK DATA name

declarations: Integer declare

Real declare

Dimension

Common [*COMDECK]

Equivalence

Data

Data--implied do

epilogue: END

Type Input/Output: An Input/Output routine may contain any type 2 statements, plus the following:

BACKSPACE u

ENDFILE u

READ(u,k) [l]

WRITE(u,k) [l]

REWIND u

READ(u) [1]

WRITE(u) [1]

FORMAT(ugh_1,ugh_2,...,ugh_n)

Type 3 Programs: are written in the assembly language of the host machine.

VII. AUXILIARY PROCESSES

VII.1.0 Input and Output of Chinese Characters

Work in improving the input and output of Chinese characters was continued in this period, with emphasis on the ability to code characters efficiently for input into SAS. The input phase was greatly helped by use of the Chinese Teleprinter System Model 600D. Output of characters still made use of the Kuno character vectors for plotting on the Calcomp. These two aspects are discussed separately below:

VII.1.1 Input of Characters

Two modes of character input were used.

(a) Keypunching on cards material which were telecoded by humans. This was the principal mode of input during the first half of the contract. The material coded consisted of both Chinese text material on nuclear physics and new lexical entries for CHIDIC. The former was gradually shifted over to using the Model 600D. The latter, coding of dictionary entries, remained a manual telecoding and keypunching task. This was because the dictionary entry format required too complex an intermixing of telecodes with the English alphabet in the gloss field to make coding by means of the Model 600D an effective process at this period. However, work has continued in this

area to seek an efficient interface on the Teleprinter System for dictionary entries.

(b) Model 600D Teleprinter System input.

More and more of the telecoding of text material was gradually shifted over to the 600D in anticipation of successful conversion of the 600D coded material to SAS acceptable telecode. At the conclusion of this contract, programs for the conversion of Model 600D code to telecode had already been thoroughly tested and debugged on the CDC 6400 at our Computer Center. However, the copying of the paper tape from the 600D onto magnetic tape could not be accomplished economically at the Computer Center since they lack the proper high speed paper tape readers. For large scale conversion of our papertape to magnetic tape a commercial data processing service bureau was tried out. Due to the non-standard code on the papertape, the copying of the papertape to magnetic tape has not given us consistently satisfactory results. However, since the problem is an independent one of obtaining highly accurate bit by bit copying of data from one medium (papertape) onto another medium (magnetic tape), it is a process which we think will soon be satisfactorily overcome in the coming months.

A total of 307 pages of nuclear physics texts, amounting to 300,000 characters have already been punched using the Model 600D and the card punch. With the special formating

which is required to preserve the format information in the original texts, it has been our experience that a skilled operator will be able to average about 20-25 characters per minute on the 600D. As compared to manual telecoding, then keypunching on cards, and verification the total increase in speed is about 3 times more efficient using the 600D. A minor disadvantage of the 600D is that it is a highly complex mechanical system which requires considerable maintenance by a trained mechanic. Otherwise, even taking into consideration the problems we have experienced in handling non-standard code on paper tape, the system is certainly the most practical one which the Project has had an opportunity to use for large scale input of Chinese characters.

VII.1.2 Description of the Chinese Teleprinter Model 600D System

The model 600D was invented by Mr. Chung-chin Kao and manufactured by the Oki Electric Co. of Japan. It has a configuration consisting of a Chinese character keyboard, a printing unit for direct hard-copy output, a paper tape punch and reader, and a slightly modified standard teletype with a Standard English Keyboard.

There are 4,600 Chinese characters on the keyboard plus 200 other symbols consisting of punctuation, the Latin alphabet and the Chinese and Arabic numerals. The characters are

arranged by radical and stroke order. The central section of the Keyboard is occupied by 1,600 of the high frequency characters and set off visually by a different color. There are 600 keys, with 8 characters or symbols on each key. A specific character is located by pressing with the right hand the key which contains the character. The left hand presses one of 8 keys that corresponds to the location of the character with respect to its location within the key pressed by the right hand. Thus every character is located by operating with both hands.

Once the two keys are depressed, the character is punched onto paper tape using a non-standard ASCII coding scheme, requiring 5 frames on the tape to represent one character. At the same time one can optionally have the character displayed on the printer. Alternatively, a whole papertape can first be punched and then fed through the paper-tape reader for printing of the whole text. This is a significant option since under printing mode the maximum input speed is only half that of the punch only speed of 120 characters per minute. A 60 character per minute speed is often exceeded during bursts of speed by the operator in dealing with very familiar characters. The advantages of having a hard-copy capability are obvious when texts have to be verified.

In order to interface this input device with the rest of the SAS, it was necessary to first transfer the papertape

information onto magnetic tape and then converted to standard telecode before the text material can be used as input. This conceptually straightforward two-step conversion turned out to be more time-consuming than was anticipated. The central factor was that there were simply few easily accessible high-speed papertape readers available whose software could accurately transfer the data bit by bit because the paper tape was in non-standard code, thus giving rise to parity errors. Another factor was due to inconsistency in the hardware operation of these readers, which sometimes missed whole sections of punched text. These problems are gradually being overcome, as mentioned earlier.

In the following pages a sample of the text coded by the Model 600D is shown together with the original text and the telecoded output obtained from the conversion routine. The routine can print out or punch the converted telecode material in card image format, as shown here, for further inspection, or the telecoded text can be retained directly on magnetic tape for direct input into the SAS.

Note that in order to circumvent the limitation of the fixed 4,600 Chinese character set of the Model 600D, we have devised a two-mode coding system which can intermix directly Chinese characters and telecodes. Thus where a character is not available on the keyboard, a telecode is substituted. It was thus also possible to incorporate all our previous format

THE TEXT IN THE NEXT PAGE IS TAKEN FROM

受控熱核反應

（理論基礎及探索成就）

第三章　高溫等離子體動力學

盧鶴紱　周同慶　許國保等　編著

上海科學技術出版社

1962年 8月　第1版

Controlled Thermonuclear Reactions

(Theoretical Foundations and Research Achievements)

Chapter 3 High-Temperature Plasma Dynamics

edited by Lu He-fu, Zhou Tong-qing, Xu Guo-bao, et al.

Shanghai Science and Technology Publishers
first edition, August 1962

第三章　高温等离子体动力学

§1 概　　說

如所周知，根据物质整体的表观性质，人們把物质存在的形态分成固态、液态和气态，通称物质三态。此三态不过是原子或分子間的結合或凝聚程度不同的表現。当晶体中每分子的平均动能超过晶体結构的結合能

$$U_1 \sim 1\ \text{ev}$$

时，晶体結构瓦解，即形成液态，或直接形成气态。当液体中每分子的平均动能超过分子間范德瓦耳斯(Van der Waals)力的結合能

$$U_2 \sim 1\ \text{ev}$$

时，液态即轉成气态。依此类推，則当气体中粒子的平均动能超过原子或分子的电离能

$$U_3 \sim 10\ \text{ev}$$

时，則气体变成电子、离子及电中性粒子的混合气体。此电离化的气体可称为物质的第四态。温度增高，中性粒子逐漸减少，即气体的电离度增大，最后气体完全电离；再增高，甚能使原子中的所有电子全部离去，变成完全由自由电子和裸露的原子核所組成的体系。再往上推，則当粒子的平均动能超过核子在原子核中的結合能

ORIGINAL TEXT

03 0 2 1 5

02第三章0101高温等離子體動力學

02 9 9 7 S 1 0101概說

0101如所周知，根據物質整體的表現性質，人們把物質存在的形態分成固態 9 9 7 7液態00和氣態，通稱物質三態。此三態不過是原子或分子間的結合或凝聚程度不同的表00現。當晶體中每分子的平均動能超過晶體結構的結合能

00 0101 9 8 5 C U 9 8 5 T 1 9 9 6 Λ 1 9 7 7 3 9 8 5 C V

00時，晶體結構瓦解，即形成液態，或直接形成氣態。當液體中每分子的平均動能超00過分子間范德瓦斯（ 9 8 5 C V Λ N 01 D E R 01 9 8 5 C W Λ Λ L S）力的結合能

00 0101 9 8 5 C U 9 8 5 T 2 9 9 6 Λ 1 9 7 7 3 9 8 5 C V

00時，液態即轉成氣態。依此類推，則當氣體中粒子的平均動能超過原子或分子的電00離能

00 0101 9 8 5 C U 9 8 5 T 3 9 9 6 Λ 1 0 9 7 7 3 9 8 5 C V

00時，則氣體變成電子 9 9 7 7離子及電中性粒子的混合氣體。此電離化的氣體可稱為物質00的第四態。溫度增高，中性粒子逐漸減少，即氣體的電離度增大，終至氣體完全電00離；再增高，且能使原子中的所有電子全部離去，變成完全由自由電子和裸露的原00子核所組成的體系。再往上推，則當粒子的平均動能超過核子在原子核中的結合00能

TEXT CODED USING MODEL 600D CHINESE TELEPRINTER

999P 0215 999H 4574 0005 4545 9998 9998 7559 3306	0001
4583 4418 1311 7555 0520 0500 1331 999H 997S 995A	0002
9998 9998 2861 6141 9999 9998 9998 1172 2076 0719	0003
4249 9976 2704 2354 3670 6347 2419 7555 4104 5903	0004
6034 1840 6347 9976 0086 0226 2116 3670 6347 1317	0005
0961 4104 1748 1966 0433 2052 0942 1966 9977 3210	0006
1966 9999 0735 3051 1966 9976 6639 4468 3670 6347	0007
0005 1966 9975 2974 0005 1966 0008 6665 2508 0626	0008
1311 2057 0433 1311 7035 4104 4814 0678 2057 0413	0009
5112 4453 1653 0008 0681 4104 5903 9999 3807 9975	0010
3981 2533 7555 0022 3020 0433 1311 4104 1627 0971	0011
0520 5174 6389 6665 2533 7555 4814 2845 4104 4814	0012
0678 5174 9999 9998 9998 985C 985T 995A 996A 995A	0013
9773 985C 9999 2514 9976 2533 7555 4814 2845 3907	0014
6043 9976 0613 1748 2052 3210 7555 9976 2057 4160	0015
2234 1748 2052 3051 1966 9975 3981 3210 7555 0022	0016
3020 0433 1311 4104 1627 0971 0520 5174 6389 9999	0017
6665 0433 1311 7035 5400 1795 3907 2448 9988 985C	0018
9895 9874 9887 9998 9877 9878 9891 9998 985C 9896	0019
9874 9874 9885 9892 9989 0500 4104 4814 0678 5174	0020
9999 9998 9998 985C 9894 985T 995B 996A 995A 9773	0021
985C 9895 9999 2514 9976 3210 1966 0613 6567 2052	0022

TEXT CONVERTED TO TELECODES

3051 1966 9975 0181 2974 7352 2236 9976 0463 3981	0023
3051 7555 0022 4721 1131 4104 1627 0971 0520 5174	0024
6389 6665 0626 1311 2057 0433 1311 4104 7193 9999	0025
4418 5174 9999 9998 9998 985C 9894 985T 995C 996A	0026
995A 995J 9773 985C 9895 9999 2514 9976 0463 3051	0027
7555 6239 2052 7193 1311 9977 4418 1311 0644 7193	0028
0022 1840 4721 1311 4104 3236 0678 3051 7555 9975	0029
2974 7193 4418 0553 4104 3051 7555 0668 4468 3634	0030
3670 6347 9999 4104 4574 0934 1966 9975 3306 1653	0031
1073 7559 9976 0022 1840 4721 1311 6632 3362 3253	0032
1421 9976 0613 3051 7555 4104 7193 4418 1653 1073	0033
1129 9976 4807 5267 3051 7555 1346 0356 7193 9999	0034
4418 9979 0375 1073 7559 9976 0011 5174 0169 0626	0035
1311 0022 4104 2076 2589 7193 1311 0356 6752 7180	0036
0637 9976 6239 2052 1346 0356 3945 5261 3945 7193	0037
1311 0735 5953 7216 4104 0626 9999 1311 2702 2076	0038
4809 2052 4104 7555 4762 9975 0375 1766 0006 2236	0039
9976 0463 3981 4721 1311 4104 1627 0971 0520 5174	0040
6389 6665 2702 1311 0961 0626 1311 2702 0022 4104	0041
4814 0678 9999 5174	0042

codes (in telecode) and provide a smooth transition from manual coding to machine coding.

VII.2.1 Output of Characters

Although the needs of having efficient methods of character input occupied a greater proportion of our effort during this contract, we were also able to make some headway into the more efficient output of Chinese characters. The capabilities of the Computer Center's Stromberg-Carlson SC4020 microfilm plotter was investigated in terms of its ability to produce readable Chinese and English characters on the same page. As a first test twenty pages of CHIDIC was obtained on microfilm using the SC4020. These did not include any Chinese characters since it was necessary to write special routines to read in the Kuno vectors. We have since developed the routines for accessing the Kuno vectors from the Extended Core Storage of the CDC 6400. However, the system software of the SC4020, maintained by the Computer Center, still needed further development before we can successfully intermix Chinese and English characters on the same page of output.

It should be noted that, although character output is not needed in a Chinese to English MT system, the necessity of having such a capability is quite obvious when one considers the part played by concordances and dictionary entries in the work being performed by the linguist and the lexicographer. The

present use of telecode output alone is an extremely inconvenient and time-consuming method of inspection by humans who are charged with the tasks of improving the linguistic capabilities of the MT system.

VII.2.2 Calcomp Tree Plots

Extensive use was made of the existing tree plotting capabilities of the SAS to aid the linguists in diagnosing sentences which were ambiguously parsed. In earlier work, the plotted trees were often too large for effective inspection. However, since the institution of parsing units of smaller size in the system, the plots have correspondingly decreased in size and complexity and has proved to be highly effective diagnostic aids in analysis and interlingual work.

The plotting routines have been modified so that it is now possible to request for specific plots of individual sentences. Previously it was necessary to plot all sentences from any particular run. The new freedom in choice of plots results in a great saving in computer time since only sentences requiring special attention will be requested for plotting.

VIII. CONCLUSION

The statistics from the runs of texts discussed have shown a general improvement in the analytical apparatus. It shows a definite trend towards a decrease in the kinds of ambiguity encountered in our earlier work. Within the existing framework, we have seen that careful research into the properties of Chinese has yielded significant results in areas where syntactic problems could be isolated without having to appeal extensively to a great deal of as yet little understood semantic information and the even more elusive pragmatic information.

A very basic problem with the types of ambiguities we have encountered arose as a direct result of the amount of information available to each lexical item in the dictionary. The earlier efforts had to deal with more scanty information than later efforts simply because there was a limitation in terms of human effort in arriving at the correct information and reducing these to machine manageable data. This applies both to lexical entries and grammar rules. An improvement in one necessarily calls for improvement in the other. The as yet to be tackled problems are clearly reflected in the results of our runs. A great deal of ambiguity has to be resolved still in the areas of complex noun phrases and verb phrases and consequently the sentence as a whole. But note that these are

exactly the major constituents in the sentence which would require information from the semantic and pragmatic spheres before their ambiguities can be adequately resolved. The research in this type of information brings us to the very forefront of the state of the art in language analysis and contrastive Chinese-English studies. This linguistic information must be adequately applied in a programming environment suitable for its manipulation. The work in artificial intelligence research appears to present a method of capturing the semantic and pragmatic information necessary for a good MT system. It is in the light of a combination of sound linguistic analysis and the "artificial intelligence" approach to programming, especially the incorporation of heuristic processes, (for example in the works of Nilsson (1971) and Winograd (1972)) that our MT system will reap the best fruits in the near future.

REFERENCES

Brekle, Herbert E. 1970. Generative Satzsemantik und Transformationelle Syntax im System der Englischen Nominal Komposition. Wilhelm Fink Verlag. Munchen.

Dahl, O. J., E. W. Dujkstra, C. A. R. Hoare. 1972. Structured Programming, Academic Press, New York, New York.

Dijkstra, Edsger W. 1969. "Notes on Structured Programming." Report No. EWD249, Technical University, Eindhoven, The Netherlands.

Henderson, P., and R. Snowdon. 1972. "An Experiment in Structured Programming." BIT 12 (1972), 38-53.

Lakoff, George and Stanley Peters. 1969. "Phrasal Conjunction and Symmetric Predicates", in Modern Studies in English, ed. David A. Reibel and S.A. Schane, Prentice Hall, N.J.

Lees, Robert. 1970. 'Problems in the Grammatical Analysis of English Nominal Compounds". Progress in Linguistics, edited by Manfred Birwisch and Karl E. Heidalph, pp. 174-186. Mouton.

Li, Charles. 1971. Semantics and the Structure of Compounds in Chinese. Ph.D. Dissertation. University of California, Berkeley.

Nilsson, N. J. 1971. Problem Solving Methods in Artificial Intelligence, McGraw-Hill, New York, New York.

Wang, S-Y. William, Benjamin K. T'sou and Stephen W. Chan. 1971. Research in Chinese-English Machine Translation. University of California. RADC-TR-71-211, Final Report, November 1971.

Winograd, T. 1972. Understanding Natural Language, Academic Press, New York, New York.

Wirth, Niklaus. 1971. "Program Development by Stepwise Refinement." CACM 14:4 (April, 1971), 211-227.

Zimmer, Karl. 1971. "Some General Observations about Nominal Compounds". Working Papers in Language Universals, No. 5, Stanford University.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-74-22	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* DEVELOPMENT OF CHINESE-ENGLISH MACHINE TRANSLATION SYSTEM		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 1 Sep 70 - 30 Dec 72 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Dr. William S-Y Wang Mr. Stephen W. Chan		8. CONTRACT OR GRANT NUMBER(s) F30602-71-C-0116
9. PERFORMING ORGANIZATION NAME AND ADDRESS The University of California at Berkeley 118 California Hall Berkeley, CA 94720		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 4594 08 01
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRDT) Griffiss Air Force Base, New York 13441		12. REPORT DATE February 1974 13. NUMBER OF PAGES 142
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*) Same		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

Same

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Computational Linguistics
Chinese-English Machine Translation
Structural/Descriptive Linguistics
Automated Lexicography
Computer Programming

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The report documents progress and results of a 2-1/3 year effort to further develop the prototype Chinese-English Machine Translation System. Additional rules were incorporated into the existing grammer for Chinese analysis and interlingual transfer, with emphasis on the latter. CHIDIC was updated and revised. Approximately 16,000 new entries were added to CHIDIC, bringing the total available entries to over 73,000. Linguistic work on a random access dictionary incorporating feature notation was carried out. A new design for the translation

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

20. ABSTRACT CONT'D

system was initiated and partially programmed for conversion of the current system from a CDC 6400 version into an IBM version. Better control of the parsing process was achieved by improving the segmentation procedures during input, and by addition of more revealing diagnostic printouts as steps toward reduction of spurious ambiguities. The Model 600D Chinese Teleprinter System was used for the first time to prepare large batches of texts for input. A total of 307 pages of machine readable texts, comprising 300,000 characters were prepared during this report.

MISSION
of
Rome Air Development Center

RADC is the principal AFSC organization charged with planning and executing the USAF exploratory and advanced development programs for electromagnetic intelligence techniques, reliability and compatibility techniques for electronic systems, electromagnetic transmission and reception, ground based surveillance, ground communications, information displays and information processing. This Center provides technical or management assistance in support of studies, analyses, development planning activities, acquisition, test, evaluation, modification, and operation of aerospace systems and related equipment.