[*Mechanical Translation*, vol.5, no.3, December 1958; pp. 114-120]

German Sentence Recognition †

G. H. Matthews and Syrell Rogovin, Massachusetts Institute of Technology, Cambridge, Massachusetts

A computer program is described which assigns one or more distinct immediate constituent analyses to every German sentence, thus indicating which of all possible sentences any given sequence of words may represent, and revealing all the information implicitly or explicitly contained in each of these sentences, that can be used in the choice of their translations.

THIS PAPER describes a routine that is based upon a theory of language which recognizes in each sentence of a given language an immediate constituent structure. Prior work on German sentence recognition [1, 2, 3] has been based on a linear view of language. Oswald and Fletcher, for example, "... found that the elements of the language in question and their functional relationships to each other could be treated most efficiently in terms of traditional descriptive grammar." [4] This theory of language that neither explains nor accounts for any features of language other than its linear structure has led them and other investigators to develop routines which merely rearrange lexical items and translate them individually into the output language.

Our general method of translation is based on the following assumptions: each sentence of a language has one or more discoverable constituent structures; there is a finite and manageable number of constructions that make up any given sentence; and these constructions, except when two or more share a single common element, are discrete from one another. Our attack on the problem of recognition has been to take one construction at a time, and develop a routine for finding its limits in any sentence, discovering that it is this construction, and finding its function in the larger construction of which it is a part. In such a program, then, difficulties do not arise from the length of a sentence, nor from the number or kinds of relationships, both syntactic and special, of its constructions; the constructions of the sentence are recognized one at a time, from the most inclusive to the least, and from the beginning of the sentence to the end. We feel that the most efficient program and the best output text can be attained by working from the outset from grammars of the two languages involved. These grammars are adapted for the computer from the type suggested by Noam Chomsky in Syntactic Structures. [5] Each grammar is a series of ordered, completely unambiguous rules, some of which are obligatory, and some, optional. Every sentence in the language is thus the result of all the applicable obligatory rules plus none or more of the applicable optional ones. Then, when the computer, as a final step in the translation process, is given the grammar of English and directs which optional rules of the grammar are to be chosen, the sentence so designated will be generated. Preceding this there is a routine which will translate lexical items, the syntactic functions of which will have been defined in the preceding step, the recognition routine. In the recognition routine the input sentence is sent through a program which ascertains those rules of gram-

† This work was supported in part by the U.S. Army (Signal Corps), the U.S. Air Force (Office of Scientific Research, Air Research and Development Command), and the U.S.Navy (Office of Naval Research); and in part by the National Science Foundation.

1. Oswald and Fletcher, "Proposals for the Mechanical Resolution of German Syntax Patterns", Modern Language Forum, Vol. XXXVI, no. 3-4 (1951).

2. Booth, Cleave and Brandwood, Mechanical Resolution of Linguistic Problems. (London, 1958) pp. 125-277.

3. Leonard Brandwood, "Some Problems in the Mechanical Translation of German", MT, Vol.V, no. 2, pp. 60-66.

4. Oswald and Fletcher, op. cit. pp. 2-3.

5. Noam Chomsky, Syntactic Structures, 'S-Gravenhage, The Hague, 1957.

mar which must have been applied in order to produce that particular sentence. The middle step, therefore, is not merely the translation of lexical items but also a translation of rules. In order to discover what rules of the grammar of the input language produced the sentence to be translated, we need, of course, a grammar of that language as well. We can thus outline the process of translation in three steps, "recognition of the structure of the incoming text in terms of a structural specifier; transfer of this specifier into a structural specifier in the other language; and construction to order of the output text specified."[6]

The authors believe that this system has advantages over those previously proposed. One feature which may appear to be a drawback is the fact that in addition to lexical translation, the detailed grammars described in the last paragraph must also be drawn up. The project is thus of necessity long range, the goal being to develop a program which will translate most effectively, rather than as effectively as possible after a short amount of time devoted to basic research. Furthermore, by basing the program on the theory that sentences are generated and thus have a traceable history, we can produce a superior output text.

It may be noted that the initial research required for our program may entail more work than that necessary for word-for-word programs but the generation of English sentences as a result of translation from any language at all will remain the same. Similarly, the recognition of German sentences will also remain constant as the first step for translation into any language. Thus two of the three sections of the program are not uniquely adapted to a particular pair of languages. If, however, the process of recognition, translation, and construction were integrated in a translation routine, the entire program would have to be unique for each pair of languages and no part of the program could be used in any other program. It is certainly reasonable to assume that we will eventually want to translate material to and from several languages. . It is therefore practical to develop a program which is not completely unique but one that has parts that can be used repeatedly, just as, within the program itself, we will want to build sections which can be used at several points in the program.

For the foregoing reasons the M. I. T. mechanical translation group has chosen to design the kind of translation program de scribed by Yngve.[6] The first step in such a program is a recognition routine; the one which we have designed is one type to come out of the approach we use. This does not preclude the possibility of others, some of which are already under investigation.

Problems of Recognition

Recognizing a sentence involves the discovering of the possible phrase structures that can be assigned to the sentence, as well as the particular morphemes used. Complicated as the generation of sentences in a natural language is, the recognition of those sentences is even more complex. The recognition process must take into account generation rules which delete, rearrange, expand, and reclassify constituents in the sentence. Further, recognition does not necessarily end when a single structure for a given sentence has been discovered, for a sentence in isolation may represent several structures, any one of which might be the "correct" one in the larger context from which the sentence was taken. The program described in this paper attempts to discover all possible structures for each sentence but obviously cannot decide which is the correct one.[7] Problems of multiple meaning have been discussed in several publications with various methods of solution proposed.[8, 9, 10, 11, 12, 13] One possible way, is that of looking at the context of one or two words before and after the word in question, but this is extremely time consuming. If it is possible to recognize the constituent structure, however,

6. V. H. Yngve, "A Framework for Syntactic Translation", MT, Vol. IV, no. 3, pp. 59-65.

7. Robert B. Lees, "Review of Noam Chomskys Syntactic Structures", Language, Vol. 33, p. 406 (1957).

8. Abraham Kaplan, "An Experimental Study of Ambiguity", MT, Vol. II, no. 2, pp. 39-46.

9. A. Koutsoudas and R. Korfhage, "Mechanical Translation and the Problem of Multiple Meaning", MT, Vol. III, no. 2, pp. 46-51.

10. Roderick Gould, "Multiple Correspondence", MT. Vol. VI, no. 1/2, pp. 14-27.

11. M. M. Masterman, "Thesaurus in Syntax and Semantics", MT, Vol. IV, no. 1/2, pp. 35-44.

12. Kenneth E. Harper, "Semantic Ambiguity", MT, Vol. IV, no. 3, pp. 68-69.

13. Kenneth E. Harper, "Contextual Analysis", MT, Vol. IV, no, 3, pp. 70-75.

then phonemically identical forms which belong to different form classes, such as gut and Gut will automatically be differentiated. However, wherever two phonemically identical forms belong to the same form class, such as Band = volume, Band = ribbon, and Band = bond, it is best to put off the solution until after the constituent structure has been determined, for it will then clearly designate just what the context is, and thus replace the ad hoc definition of context, which is used in the above cited papers.

Operation of the Routine

The routine itself is divided into several parts - initialization, dictionary search, determination of the kind of sentence that is being recognized (i.e. is it a question, declarative sentence, if-then construction, etc.), delimiting subordinate constructions and removing them from the main clause, establishing the limits and possible functions of the several noun phrases in the sentence, and determining what verb forms are present and what their governance relationships are. Finally the actual functions of the noun phrases are determined. After this operation has been performed on the main clause, the process is repeated for each dependent construction and indications are inserted concerning the use each construction has either in the main clause or in another dependent construction.

Initialization

Initialization involves bringing the sentence letter-by-letter into the workspace. ('Workspace' is the designation in the M.I. T. programming language, [14] for an expansible register in which strings of symbols are manipulated.) Each symbol is tested to see whether it is a space between words (space is treated as an orthographic symbol), in which case the sequence between it and the last such space is placed at the beginning of the workspace so that at the end of the initialization process the words are in reverse order. Each character is also tested to see whether it is a terminal punctuation mark, in which case the input part of the routine has been completed. Thus the unit of translation is a complete sentence. It is probable that in a connected text information gleaned from one sentence might be useful in recognizing the structure of following sentences. Such information would be useful in choosing among several possible phrase structures or meanings. However, to date we have not incorporated this information in our program.

Search

Following the initialization words are looked up in the order in which they appear in the workspace, i.e. from the end of the sentence to the beginning. The dictionary is divided into two separate parts; the first is a list of separable prefixes in which the last word of the sentence is first looked up. A typical entry in this part of the dictionary AUF // SW1 SEP 3. This is a rule in the programming language used at M.I. T. for expressing linguistic facts in a manner that can be interpreted by a computer. This rule means that if the last word in the sentence is auf it will be found, a note will be made that of the set of alternative rules designated by SW1 the particular rule that will be chosen is rule SEP, and the next rule to be applied is rule 3. This first part of the dictionary contains an entry for every separable prefix. Later SW1 SEP will cause the finite verb of the sentence to be looked up in conjunction with the separable prefix. When wieder appears as the last word in the sentence, it may present an ambiguity, e.g. Er kommt wieder, can be either "He is coming again, " or "He is coming back," if wieder is an adverb in the first sentence and a separable prefix in the second. In cases like this, two interpretations will be offered. All other words in the sentence, as well as the last one if it is not found in the separable prefix list, are looked up in the main dictionary. The entries in this dictionary have the effect of adding grammatical information in the form of subscripts to the word that is looked up. The specific form of the entry depends mostly on the form classes to which the entry word belongs, and partly on the particular word itself. Every possible German lexical item which one would want to translate is included in the dictionary. This is feasible because storage space in the form of tapes is essentially unlimited. Our program has been written so that the dictionary must contain an entry for every form to be translated. However, if it should prove to be more efficient, a sub-routine could be added which would remove endings from a stem. The dictionary would then need to contain only one entry for each morpheme. However, due to the productiveness of compounds in German, especially in scientific literature, it would be well to have a sub-routine which would indicate and look up separately their

14. V. H. Yngve, "A Programming Language for Mechanical Translation", MT, Vol. V, no. 1, pp. 25-42.

constituents.[15] This, of course, should not be done in cases where just one of two or more possible interpretations is correct, such as Literaturkunde, or where the meanings of the compound is not the same as the sum of its constituents such as Hochzeit. It would also be well to give two interpretations to ambiguous compounds such as Bluterzeugung. Some typical entries in the lexicon are:

```
BUCH      = 1/.1, CASE -GEN, PN 3S,
            GEND NEUT, CNG 1 5 9
LIEST     = 1/VRB, CASE ACC, PN 3S,
            FORM FIN, TYPE MAIN,
            TENSE PRES
DASS      = Y4 + SB1 + 1/CON -SUB
DEN       = 1/.15, CASE ACC DAT,
            GEND MASC PLUR, CNG 6 11
GEHENDE   = 1/.25, CASE NOM ACC,
            PN 3S 3P, CNG 1 2 3 4 5 7 8,
            FORM PRES-ADJ + Y1
IN        = 1/.20, CASE ACC DAT,
            CNG 5 6 7 8 9 10 11 12
SCHWEREN  = 1/.5, PN 3S 3P,
            CNG -1 2 3 5 7
```

In each of the above subscripts, the first symbol of a set between commas names a class and the following symbols of the set are the members of the class to which the lexical item may belong. The subscripts attached to BUCH give us the following information: . 1 means the word is a noun (numerical subscripts will be discussed later); CASE -GEN means the word may be any case except genitive; PN 3S indicates its person-number qualification is third singular; GEND NEUT shows it is in the neuter gender; (plural is also regarded as a gender); CNG stands for a coding which combines case, number, and gender in a two-dimensional scheme which shows number-gender horizontally in the order, neuter, masculine, feminine, plural and case vertically in this order: nominative, accusative, dative, genitive. Numbering begins at the upper left and moves horizontally.

For entry LIEST, CASE ACC means that the verb takes an object in the accusative case. FORM includes finite, infinitive, past participle, participle with an adjectival ending. TYPE is main, auxiliary, passive, modal or future.

If a word is not found in the lexicon, the sentence is automatically printed out and that word is letter-spaced. This would happen most often in the case of proper names. An alternative procedure would be to have a pre-routine which would merely look up all the words of the text in the lexicon, printing out those which are not found. Then, when entries for these forms had been made in the dictionary, the recognition program could proceed.

15. Erwin Reifler, "Mechanical Determination of the Constituents of German Substantive Compounds", MT, Vol. II, no. 1, pp. 6-14.

The Process of Recognition

Following the placement of subscripts on the lexical items, we come to the main portion of the routine. In effect it does the following: Considering the beginning of the sentence to be the left and the end to be the right, the program scans from the left looking for the finite verb. Arriving at the right, the scanner then proceeds in the other direction to locate dependent constructions, each of which is removed from the main clause, whereupon a marker is left in its place. Once more at the left, the scanner reverses its direction and moves along locating and classifying the phrases which remain.

Location of Finite Verb

We shall now examine the process of recognition in more detail: When all the forms in the German sentence have been looked up in the dictionary, their order in the workspace is reversed, so that they are now in the order of the original sentence. Then the finite verb of the main clause is located, placed at the end of the sentence, and its original position is marked. This is done in order to connect the verb stem with a possible separable prefix. The finite verb form of the main clause is moved so that all clauses, dependent and independent, may be treated alike by the rules which follow. We now come to the previously discussed set of rules, SW1. If rule SEP has been indicated, the last two elements in the workspace, i.e. the separable prefix and the finite verb, will be looked up again in the dictionary, and a different set of grammatical information will be assigned to it.

AUF-STEIGT = 1/VRB, etc.

The following are the rules for determining the finite verb: 1) In sentences containing a single clause, the finite verb is the first verb in the sentence which can be finite. 2) In complex sentences where the dependent clause precedes the finite verb of the main clause, we require that the dependent clause be followed by a comma and that each such relative clause which does not begin the sentence be preceded by a comma. Assuming that these requirements are met, we choose as the finite verb of the main clause the first finite verb-form of the sentence which is not within a dependent clause. 3) Sentences

which fall into neither of the above categories (e.g., with final dependent clauses), can be treated under the first rule.

Dependent Constructions

The next part of the routine establishes the limits of the dependent constructions — subordinate clauses, relative clauses, and participial phrases — and places them at the beginning of the workspace in the same order in which they occurred in the sentence. In establishing the limits of these constructions, those which are nested within other dependent constructions are, for the time being, ignored and are automatically moved to the beginning of the workspace with the constructions in which they are embedded. The general method of discovering these limits is to work from the end of the sentence and to place a right parenthesis, so to speak, at the end of each such construction and a left parenthesis at the beginning. Whenever the number of lefts equals the number of rights, the leftmost and the rightmost are the limits and every thing between them is moved to the beginning of the workspace. This process is repeated until the beginning of the sentence is reached. Whenever a dependent construction is moved to the left of the workspace, an indication of it is inserted in its original position, and it is separated from other constructions by special marks.

The criteria applied in placing these parentheses are: a right is placed 1) after each sequence of a finite verb plus a punctuation mark and 2) after each participial form with an adjectival ending. A left is placed 1) before a subordinate conjunction and any punctuation that precedes it, 2) in the case of a participial construction, between any constituent of a prepositional or noun phrase and a word which could not be a constituent of the same phrase, and 3) in the case of relative clauses, before an unambiguous relative pronoun or before a sequence of comma (or comma plus preposition) plus a definite article which is in turn followed by a word which could not be part of the same construction as the article. In the case of transitive participles the program recognizes the fact that the noun preceding the participle is part of the participial construction. Thus, in ein Leben spendendes Weib, the left parenthesis is placed after ein.

Identification of Phrases

At this point, the main clause of the sentence is at the end of the workspace and a mark has been placed at its beginning. The next part of the program is designed to delimit the several noun phrases and prepositional phrases and to establish their possible functions.

Since the dictionary entries attach code numbers to prepositions and all constituents of noun phrases — prepositions, articles, numerals, adjectives, and nouns, numbered from highest to lowest, respectively, — the program accomplishes the first of these operations by scanning the workspace comparing the numbers and whereever there is a sequence of one number followed by a higher number, an equal number which is not the adjective number, or by no number at all, that point is regarded as the end of the phrase, the grammatical information previously attached to each element by the dictionary is compared in order to find the possible functions of this construction in any German sentence.

DER/. 15, CASE -ACC, GEND -NEUT,
CNG 2 7 8 11

GUTE/. 5, CASE NOM ACC, PN 3S,
CNG 1 2 3 4 5 7 8

MANN/. 1, CASE -GEN, PN 3S, GEND MASC,
CNG 2 6 10

The grammatical information associated with the words of this noun phrase is compared by an automatic process akin to taking a logical product. The results of this are indicated at the beginning of the phrase on a marker Y4, Y4/. 1, CASE NOM, PN 3S, GEND MASC, CNG 2 This process is repeated for all phrases in the clause, and the markers then represent the gram matical meaning of each of them. In the case of a prepositional phrase, the grammatical information attached to the preposition is compared with that of the elements of the noun phrase to discover its function in the sentence.

Following this, the verbal elements of the clause are considered. The purpose of this portion of the routine is to recognize what verbal elements occur, what their relationship to each other is, and to place an indicator at the end of the clause to represent the grammatical meaning of each of these forms. In the case of ambiguous verb sequences such as Das Kind wird vergessen, if selection rules allow the noun phrase to be both a subject or an object of the main verb in its active voice, the program will first designate the sequence as both passive and future and in a later part of the program it will provide two constituent analyses, one passive and one future, each of which is represented by the sequence of words in the sentence.

Assignment of Syntactic Functions

The program next assigns syntactic functions to the several noun phrases. In general, the criteria for choosing which of the noun phrases is the subject are the same as those outlined by Oswald and Fletcher[16] and by Brandwood.[17]

The program is here divided into three sections, one to treat each of three types of sentences, — passive sentences, active sentences which take accusative objects, and all others. In passive sentences, if the main verb takes an accusative object, the first possible nominative that agrees with the finite verb in person and number is regarded as the subject. In other passive sentences the first noun phrase which is of the case that the main verb would take as its object in the active voice is marked as the subject. In active clauses in which the main verb does not take an accusative object, the first nominative that agrees in person and number with the finite verb is marked as subject; if there is no such nominative noun phrase, the first dative noun phrase is marked subject. In active clauses with verbs that take an accusative, if there is an unambiguous nominative it is designated as subject; otherwise the first possible nominative noun phrase that agrees with the finite verb is designated the subject. (By a very simple addition to the program, a sentence which has two noun phrases, both of which fulfill all the grammatical qualifications for subject and object, could be printed out twice with a different assignment of subject and object in each case). The object of all active clauses is the first noun phrase that has the case required by the main verb and has not been designated subject. Noun phrases that can be either genitive or dative and which follow another noun phrase are designated genitive; other such noun phrases are designated dative.

The recognition of the main clause of the sentence is now complete. The workspace now contains the dependent constructions in the same order in which they occurred in the original German sentence but separated from the main clause and placed, with indication of their limits, in front of the main clause. However, dependent constructions that are embedded within other dependent constructions are not so separated. Following the string of dependent constructions is the main clause with one change in order, i. e. the finite verb has been placed at the end of the clause and combined with a possible preceding separable prefix.

In addition to the original words of the main clause, each with its respective grammatical information, there are also several markers, each indicating the syntactic function of the following noun phrase or the preceding verbal elements. There is also a marker which shows the original position of the finite verb, and there are indicators in the original positions of each of the dependent constructions.

The program now turns its attention to the dependent constructions. Starting at the leftmost construction it goes through the routine described above and then places that construction in the main clause in the position of the first indicator that follows it. In the recognition of a dependent construction, constructions which are in turn dependent on it are treated according to the general rule, i. e. they are placed at the beginning of the workspace and indicators are put in their places in the sentence. Thus, if the leftmost dependent construction is always taken as the next to be recognized, and upon having been recognized is placed in the position of the first indicator which follows it, all of the dependent constructions will be returned to the same place from which they were taken. In the case of participial constructions it is necessary to insert a coded symbol to function in the routine as a subject and, in the case of past participial constructions, one to function as a finite verb — auxiliary after intransitive participles and passive after transitive participles — so that the rules will apply correctly. These symbols are removed when the recognition of the construction has been completed.

The foregoing is a description of an actual program which is written in the M.I.T. programming language, a language that is being adapted for an IBM 704 computer. The authors do not claim that this program can recognize all German sentences. There are orthographic restrictions as well as grammatical ones which must be observed in order that a sentence be recognizable by this routine. An example of the former is the fact that adjectives in a series must not be separated by commas. Grammatical difficulties arise with such sentences as: "Gesprochen werden können die Worte eines Satzes. .. " or "Gehen können wir nicht. " In both cases, our program would fail to find the finite verb. These limitations on the usefulness of the routine are, however, far from disheartening. Inspecting the program one readily finds the appropriate points at which to build in a

16. op. cit., pp. 10-13.

17. Booth, Cleave and Brandwood, op. cit. pp. 161-182.

sub-routine to recognize constructions that are not at present included. The limitations do not represent an inherent weakness in the system. Rather they exemplify the results of optional transformations which we have not yet treated.