A DIRECTED RANDOM PARAGRAPH GENERATOR

Stanley Y.W. Su & Kenneth E. Harper
(The RAND Corporation, Santa Monica, California)

1. INTRODUCTION

The work described in the present paper represents a combination of two widely different approaches to the study of language. The first of these, the automatic generation of sentences by computer, is recent and highly specialized: Yngve (1962), Sakai and Nagao (1965), Arsent'eva (1965), Lomkovskaja (1965), Friedman (1967), and Harper (1967) have applied a sentence generator to the study of syntactic and semantic problems of the level of the (isolated) sentence. The second, the study of units of discourse larger than the sentence, is as old as rhetoric, and extremely broad in scope; it includes, in one way or another, such diverse fields as beyond-the sentence analysis (cf. Hendricks, 1967) and the linguistic study of literary texts (Bailey, 1968, 53-76). The present study is an application of the technique of sentence generation to an analysis of the paragraph; the latter is seen as a unit of discourse composed of lower-level units (sentences), and characterized by some kind of structure. To repeat: the object of our investigation is the paragraph; the technique is analysis by synthesis, i.e. via the automatic generation of strings of sentences that possess the properties of paragraphs.

Harper's earlier sentence generation program differed from other versions in its use of data on lexical co-occurrence and word behavior, both obtained from machine analysis of written text. These data are incorporated with some modifications in a new program designed to produce strings of sentences that possess the properties of coherence and development found in "real" discourse. (The actual goal is the production of isolated paragraphs, not an extended discourse.) In essence the program is designed (i) to generate an initial sentence; (ii) to "inspect" the result in order to determine strategies for producing the following sentence; (iii) to build a second sentence, making use of one of these strategies, and employing, in addition, such criteria of cohesion as lexical class recurrence, substitution, anaphora, and synonymy; (iv) to continue the process for a prescribed number of sentences, observing both the general strategic principles and the lexical context. Analysis of the output will lead to modification of the input materials, and the cycle will be repeated.

This paper describes the implementations of these ideas, and discusses the theoretical implications of the paragraph generator. First we give a description of the language materials on which the generator operates. The next section deals with a program which converts the language data into tables with associative links to minimize the storage requirement and access time. Section 4 describes: (1) the function of the main components of the generation program, (2) the generation algorithm. Section 5 describes the implementation of some linguistic assumptions about semantic and structural connections in a discourse.

Table 1

GOVERNING PROBABILITIES

Governor	Type of Dependent					
	VT	VI	N	A	DV	DS
VT	0	0	S 0 P_1 1	0	P_2	P_3
VI	0	0	S 0 1 0	0	P_4	P_5
N	0	0	P_6	P_7	0	0
A	0	0	0	0	0	0
DV	0	0	0	0	0	0
DS	0	0	1	0	0	0

The governing probabilities for a word are independent of each other. In paragraph generation the decision to select a dependent type will be made without regard to the selection of other dependent types. For example, a noun can have probabilities P_6 and P_7 of being the governor of a noun and an adjective respectively. The selection of a noun as a dependent based on P_6 will not affect, and will not be affected by, the selection of an adjective as a dependent.

There are two types of co-occurrence data accompanying every word in the glossary: a set of governing probabilities and a list of dependents. The probability values associated with a word are determined on the basis of the syntactic behavior of the word in the processed text. If a noun occurs in 75 instances as the governor of an adjective in 100 occurrences in a text, the probability of having an adjective as a dependent is 0.75. The zeroes and ones in Table 1 are constant for all words in the glossary. These values are not listed in the sets of probability values for the entries of the glossary; however, they are known to the system. For instance, the set of probability values for a transitive verb will contain P_1, P_2, and P_3. The probability 1 of governing a noun as object will not be listed in the data.

The second type of co-occurrence data accompanying every word in the glossary is a list of possible dependents. The list is specified in terms of word numbers and semantic classes (to be described later). It contains the words that actually appear in the processed physics text as dependents of the word with which the list is associated. Since the lists of dependents are compiled on the basis of word co-occurrence in the text, legitimate word combinations are guaranteed. In the list of dependents for a verb, those words which can only be the subject are marked "S" and those which can only be the direct object are marked "O".

The co-occurrence data can be regarded as either syntactic or semantic. They are distinguished here from both the dependency rules and part of speech designation, and from the semantic classes that have been established. At present, seventy-four semantic classes have been set up. Some of these are formed distributionally (i.e., on the basis of their tendency to co-occur syntactically with the same words in text—cf. Harper, 1965); other classes contain words of the same root, synonyms, hypernyms, and words arbitrarily classified as "concrete." The semantic classifications are highly tentative, and are subject to modification. Their extent is shown in Table 2.

Table 2

SEMANTIC DATA

Classification	Number of Classes	Number of Words in Class
Distributional Classes	22	150
Hypernym Classes	10	160
Word Families	25	52
Synonym-antonym Classes	16	48
"Concrete" Words	1	54
TOTAL	74	464

The language materials described above are punched on approximately 2500 cards. The data are processed by a conversion program in order to form the data base for the paragraph generation program.

3. DATA CONVERSION PROGRAM

The paragraph generator is written in Pl/1 and run on the IBM 360 Model 65. It consists of two main programs: a data conversion program and a generation program. These two programs run as separate jobs. The data conversion program converts the language materials described above into compact data tables with associative links. The converted data are stored on a magnetic tape which is used as input to the generation program.

During the process of paragraph generation it is desirable that the language data described in the preceding section remain in core storage. However, since the data base is rather large, its conversion into a more compact and flexible form is desirable so that storage requirements and access time can be reduced.

In view of the characteristics of the language data and the generation algorithm (to be described later), we structure the data base in the following way.

a. Words belonging to the same parts of speech or semantic classes are stored in consecutive locations. In the process of generation, random selection of a word from a part of speech or a semantic class is often required. If words are grouped together in form of a table, a randomly selected number in a proper range can be used as an index to look up the word from the table.

b. The data storage for each entry of the glossary is of variable length, since the lists of dependents, governing probabilities, hypernyms and semantic classes associated with the entries are of variable length.

c. Word numbers in the lists of dependents and semantic classes are replaced by pointers, which identify the locations where the word numbers are actually stored. Thus, data tables containing different types of information are linked to one another, and access to this information can be carried out by straight table lookup.

In keeping with these principles, data tables of the form shown in Fig. 1 have been constructed. An example will illustrate the organization of the data base and the procedure of setting up these data tables. As a noun, represented by word number 2466, and its associated language data are read from the input unit, the word number is stored in the block in table (1) reserved for nouns. The last three digits of the word number are used as an index to a location in the lookup table (2), where the word number 2466 and its address in table (1), i.e. 309, are stored. If the location in table (2) has been occupied (when more than one word has the same last three digits), the word number and its address are stored at the first unused space in table (2) following that location. Table (2) allows us to replace word numbers with their addresses after all data have been processed.

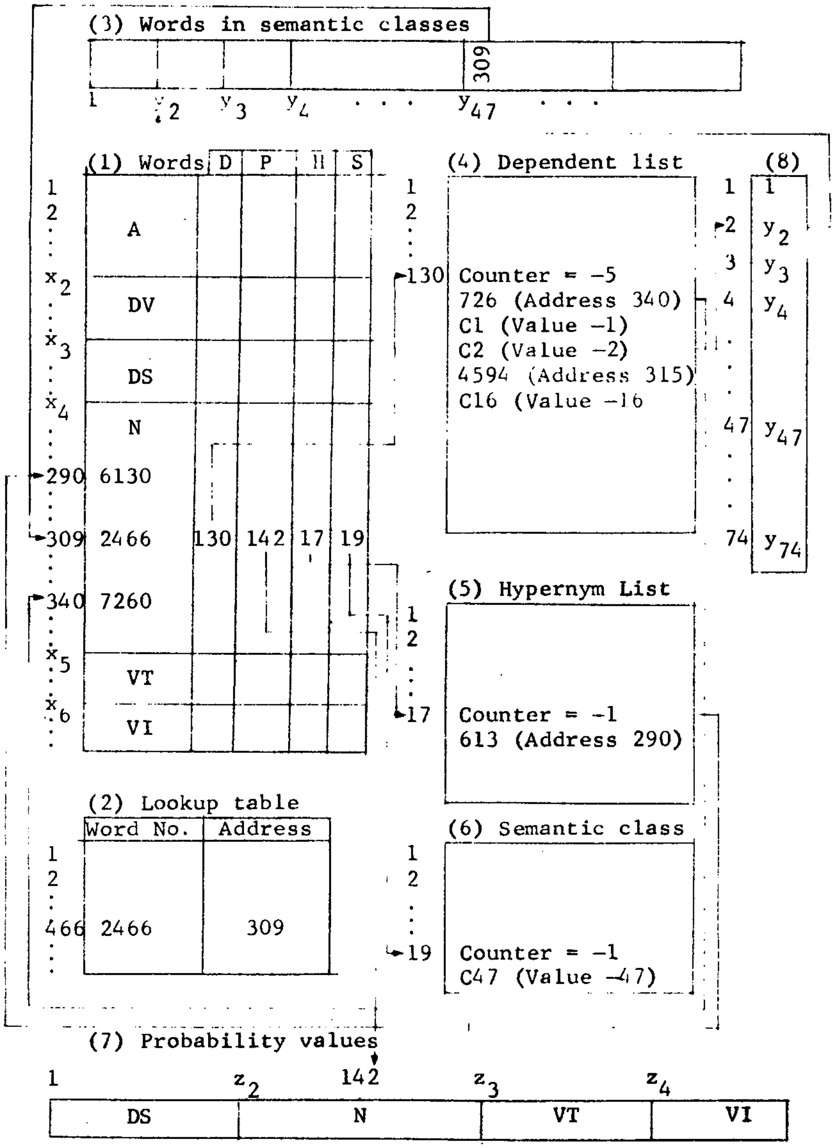


Fig. 1—Data table organization

There are four pointers associated with each word stored in table (1). Pointer D specifies the location in table (4) where the list of dependents associated with the word is stored. A counter is used to specify the number of words and semantic classes in the list. A semantic class in the original data is prefixed by a C (Cl identifies semantic class 1). In table (4) all the counters and semantic classes (the numerical values) are stored as negative values so that the positive values (i.e. word numbers) can be conveniently changed to pointers at a later stage. In our example the pointer D is 130 and the words 726 and 4594, and also the semantic classes Cl, C2 and C16, are in the dependent list associated with word 2466. The value which identifies a semantic class in table (4) is actually a pointer to a table which contains the starting locations of the lists of words in all semantic classes. This is illustrated in Fig. 1 by the links from table (4) through table (8) to table (3).

The set of governing probabilities associated with word 2466 is stored in table (7). Pointer P specifies the starting location where the probability values are stored. In the example, P is set to 142. Notice that no spaces are reserved for adjectives and adverbs because they do not have governing probabilities.

The pointer H associated with a word in table (1) specifies the location in table (5) where a counter and the hypernyms of the word are stored. Word 613 is a hypernym of the word 2466. Thus, H is set to 17 which is the location in table (5) where a counter and the word 613 are stored. Since the word 2466 is a member of the semantic class C47, the pointer S associated with the word 2466 is set to the location in table (6) where a counter and C47 is stored.

Table (3) contains 74 blocks, which are reserved for the 74 semantic classes established in the system. Each block contains a counter and the addresses of the words in a semantic class. For example, the address 309 is stored in the 47th block in table (3). Table (3) is thus linked to table (1).

After all data have been entered in the tables, the word numbers (positive values) in tables (4) and (5) are replaced by their addresses in table (1). This operation is done by using the lookup table (2).

The data are organized in tables with associative links. All word numbers in tables (3), (4), and (5) are replaced by their addresses in table (1). From an entry in table (1) (where the generation of a sentence usually begins), we can trace its possible dependents; since these dependents are specified as pointers to their addresses in table (1), it is simple to obtain the lists of dependents associated with these dependents. In turn we can trace thir level dependent lists. We can easily continue this operatio down to any desired level. Table (1) is linked to tables (4), (5), (6), and (7), and tables (4), (5), and (6) are linked either directly back to table (1) or indirectly through table (8) and then table (3) back to table (1). Thus, access to any piece of information in these data tables is gained by simple table lookup.

In view of the variability in the number of words in each part-of-speech and semantic class, and in the number of governing probabilities, hypernyms, semantic classes and dependents associated with each word, we have packed these data in large arrays as illustrated in tables (1), (3), (4), (5), (6), and (7). The advantages are (i) reduction in storage requirements, and (ii) capacity for rapid selection of a word from a part of speech or a semantic class. The disadvantage is that we have placed a restriction on the amount of additional data that may be added to the existing lists. To avoid modifying the program when new data are added, indices (such as x, y, and z in Fig. 1) to the reserved spaces in tables (1), (3), and (7) are made input parameters to the program. At present the parameters are set to leave space for expansion of input data. Further expansion can be handled simply by readjusting the input parameters.

The restriction pattern in Fig. 2 specifies that the sentence to be generated should contain a transitive verb which belongs to either semantic class C1 or C2. The verb should govern (1) a noun as the subject of the sentence, (2) an object which is to be selected from the words in semantic class C15 or the specified words W_1 and W_2, and (3) an adverb which does not belong to semantic class C19. The subject of the sentence should not govern an adjective.

As illustrated in the pattern, each node in a pattern contains a word class and selection restrictions which are positively or negatively specified in terms of semantic class(es), specific word(s) or a word class. Restriction patterns are stored in the following form: $Q{-}P_1P_2\ldots P_n$. Q is a single pattern, or a combination of patterns, and $P_1P_2\ldots P_n$ are single restriction patterns. Essentially, $Q{-}P_1P_2\ldots P_n$ is a rule which specifies that if a sentence (or string of sentences) whose sentence skeleton(s) matches Q, then it can be followed by a sentence whose sentence skeleton is one of these Ps. Thus, one of these Ps is randomly selected to be used as a restriction pattern for a succeeding sentence. The pattern selection procedure is not yet coded. At present, strings of restriction patterns are given directly to the pattern selection routine. The generation program generates strings of sentences under the control or direction of the restrictions specified in the patterns. The use of restriction patterns to control the general "development" of paragraphs will be described in a later section.

4.1.3. Discourse Relator (RELATOR). Input to this procedure are (1) a dependent type, (2) a probability value, and (3) a restriction pattern. This procedure determines whether the given dependent type conflicts with the restrictions specified in the pattern. If no conflict is found, this procedure determines whether a word should be selected from the given dependent type based on the input probability value. If the selection of the dependent type conflicts with the restriction pattern, or if the dependent type fails the probability test, no word will be selected from the dependent type.

4.1.4. (CRITERIA). Whenever, during any stage of sentence generation, the selection of one word from a list of candidates is required, this procedure determines which criteria should be applied to control the selection. All criteria (implemented principles of cohesion to be described in a later section) are presented to the generation program in the form of a table that reweights the probabilities. The generation program increases or decreases the probability of selecting words on the basis of the values in this table. It has the following format (Fig. 3): each entry is specified by a semantic class or a specific word followed by a positive or negative value.

Identifier	Weight
C2	+5
W_1	-7
W_2	+3
C12	-4
.	.
.	.
.	.

Fig. 3 — Format of a reweighting table

To illustrate the use of this table, let us suppose that in a certain stage of generating a sentence there are five words, each of which can be the subject of the verb previously selected for the sentence. The selection of any word from these five will satisfy the restriction pattern for the sentence. Instead of randomly selecting one word out of these five candidates, we may want to increase the probability of selecting a word which will have semantic connections with the word(s) in the preceding or current sentence. When there are choices in word selection, all candidates are preassigned equal weights, and criteria relevant to the current selection are applied to form a reweighting table. If a word in the list of candidates matches a word or belongs to a semantic class in the table, the associated weight is added to its preassigned weight. The final positive weights of all candidates are added, and a random number in the range from 1 to the total sum is generated to determine which candidate should be selected.

4.1.5. Word Generator (WOR_GEN). This procedure finds all possible candidates which satisfy the restrictions specified in a restriction pattern, and assigns different weights to them on the basis of the contents of a probability reweighting table. It selects a word at random from the candidates according to their weights.

4.1.6. Random Number Generator (RANDOM). Input to this procedure is an integer N. This procedure generates a random integer in the range from 1 to N.

4.2. The Generation Algorithm

The general strategy for generating a paragraph is, first, to generate the initial sentence based on a selected restriction pattern, and then to generate each noninitial sentence base not only on a selected restriction pattern but also on the semantic properties of the words in all the previously generated sentences of the paragraph. The algorithm and the sentence generation procedure can best be illustrated by an example. Let us suppose that the restriction pattern shown in Fig. 4(a) is chosen for a sentence. For ease of reference we will letter the steps involved in this procedure.

a. If the restriction pattern specifies a restriction on the selection of the sentence governor (usually a transitive verb (VT) or an intransitive verb (VI)), a VT or VI will be randomly chosen from the specified semantic class(es) or word(s). Otherwise a VT or VI will be randomly chosen. In our example the restriction pattern in Fig. 4(a) specifies that a word should be selected from the word class VT which is not a member of the semantic classes C1, C2, and C3, but is a governor of a word in C16, a word in word class N. and a word in C19. (Note also that the sentence should not contain a sentence adverb.) There are 16 candidates which satisfy the restrictions. They are shown in Fig. 4(b) by their addresses, at which word numbers are stored. A random number is generated in the range 1 to 16, for example, 8. Thus the eighth VT is chosen: word number 3336 whose address is 531.

b. The possible dependent types of a VT are NS (noun subject), NO (noun object), DV (adverb), and DS (sentence adverb). The probabilities (in percentages) for the word 3336 to govern words of these dependent types are, say, 65, 100, 35, and 40 respectively. The procedure RELATOR is called in order to determine whether the selection of each dependent type agrees with the restriction pattern. If the selection of a dependent type conflicts with the pattern, the dependent type is ignored, i.e., no word will be selected from this dependent type as the dependent of the verb 3336. In our example, NS, NO, and DV have passed this test.

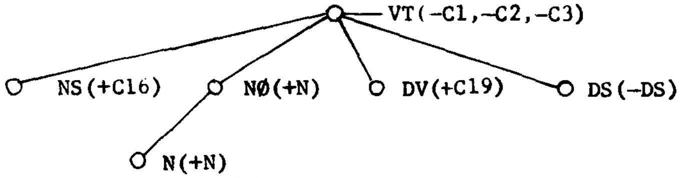


Fig. 4(a)—A restriction pattern

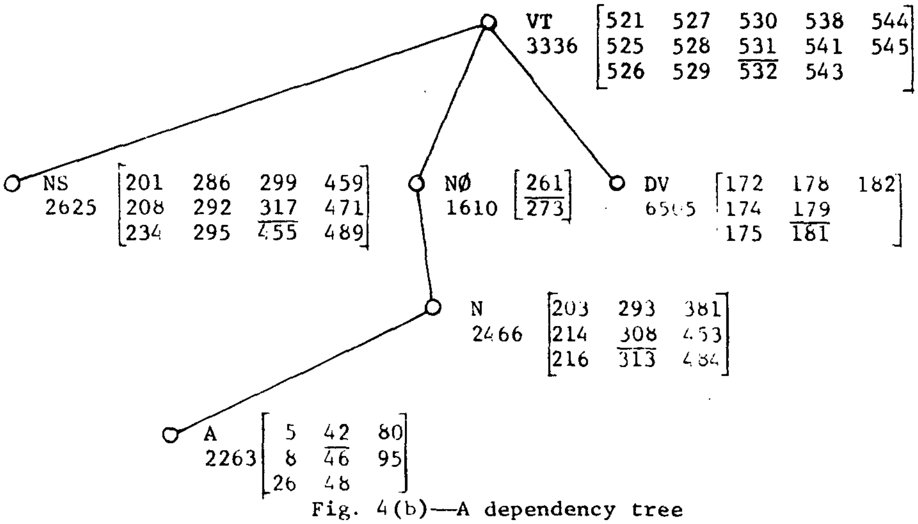


Fig. 4(b)—A dependency tree

English:	Muxin	published	a study of	linear	method	in a previous paper.		
Russian:	Muxin	opublikoval	izučenie	linejnyj	metod	v	predyduščej	rabote.
Word No.:	2625	3336	1610	2263	2466		6505	
Address:	317	531	261	42	308		179	

Fig. 4(c)—A generated sentence

c. The probabilities associated with NS, NO, and DV are used to determine whether words should be selected from these dependent types. For each dependent type, the random number generator is called to generate a number ranging from 1 to 100. If the random number is greater than the probability associated with the dependent type, the type is ignored. Otherwise a word will be selected from this type. Let us assume that all three types have passed the probability test.

d.1. A noun is to be selected as the subject of verb 3336. If the sentence to be generated is the first sentence of a paragraph, a noun which is in the dependent list associated with the verb 3336 and also a member of C16 is chosen. However, if the sentence is a noninitial one, the procedure CRITERIA is called to form a probability reweighting table on the basis of the criteria applicable to the verb 3336 and to this local structure (i.e., a VT dominates an NS). All candidates (those words which belong to C16 and which are in the dependent list associated with 3336) are first assigned an equal weight. Then the probability reweighting table is used to adjust the weights of the candidates. Fig. 4(b) shows the candidates for the node NS. An individual word is randomly chosen from the candidates based on their different weights: word number 2625 whose internal address is 317.

d.2. A noun is to be selected as the object of the verb 3336. As in d.1, the restriction pattern is consulted and, if the sentence is a noninitial one, the procedure CRITERIA is called. Fig. 4(b) shows the candidates for the node NO. The same probability reweighting scheme is applied to adjust the weights of the candidates. A word is selected at random: word number 1610 whose address is 261.

d.3. An adverb is to be selected for the verb 3336. Similar to the previous procedure, the restriction pattern restricts the selection of candidates; CRITERIA is called for a noninitial sentence to construct the probability reweighting table, and an adverb is randomly selected. In the figure we see the candidates for the node DV, and the adverb 6505, whose address is 179, is chosen.

e. The dependents of the words 2625, 1610, and 6505 are now considered with respect to their possible dependents and associated probabilities. We are working from the top to the second level of the dependency tree structure.

e.1. The noun 2625 may govern the dependent types adjective and noun. Each of these is considered in turn by the same operations described in steps b. and c. For brevity, let us assume that none of these dependent types pass the probability test. Thus, no word is selected from these dependent types.

e.2. The noun 1610 may govern the dependent types adjective and noun with different probability. Assuming that the adjective fails the probability test, none is chosen for the word 1610. Since the restriction pattern specifies that a word should be selected from the word class N as a dependent of the word 1610, the same operation described in step d. is performed to select the word 2466, whose address is 308.

e.3. The adverb 6505 selected in d.3. has no dependent since adverbs never govern.

f. We now move to the third level of the dependency tree structure. The noun 2466 may govern the dependent types adjective and noun. Let us assume that the dependent type A passes the tests described in step c. and the dependent type N fails. An adjective 2263, whose address is 42, is selected from the list of candidates shown in the figure.

g. We now move from the third level to the fourth level of the dependency tree structure. Since the only word on the fourth level is an adjective, which does not govern, we have reached the lowest level. The generation of a sentence is completed. Fig. 4(c) shows the generated sentence. (In the Russian sentence, morphology is ignored.) The restriction pattern of the sentence just generated, together with those of the previously generated sentences, are again used as the basis for selecting the restriction pattern for the next sentence.

At the present stage of development no criterion is used to determine the end of a paragraph. The number of restriction patterns input to the pattern selection procedure determines the number of sentences in a paragraph. When the sentences of a paragraph have been generated, glossary look-up is performed and the transliterated Russian forms and their structural relations are printed.

5. IMPLEMENTATION OF LINGUISTIC ASSUMPTIONS

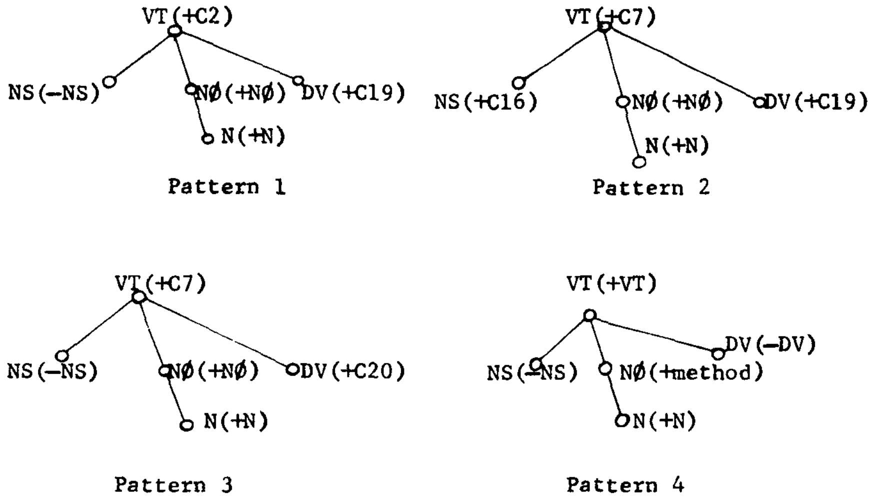
The structure of paragraphs is poorly understood, and is in any event subject to enormous variety. Nevertheless, we have adopted a simplified model, which postulates that the units (sentences) of a paragraph should be arranged in a recognizable pattern. Specifically, it is assumed that each pair of sentences should be characterized by the attributes of development and cohesion. Development implies progression—for example, some kind of spatial, temporal, or logical movement: a paragraph can be assumed to "get somewhere." Cohesion, on the other hand, implies continuity or relatedness; as such, it is a kind of curb on progression. Although it is difficult, perhaps impossible, to distinguish between these two attributes, they will be discussed separately, in an admittedly artificial way.
The chief function of the restriction pattern is to achieve intersentence development, and an overall pattern to the sequence of sentence pairs; to a degree, lexical coherence is also effected through the restriction pattern (e.g., through the recurrence of semantic classes). The main function of the probability reweighting tables is to achieve cohesion, through the device of increasing the likelihood of lexical recurrence; the principle of development is also implemented here, to the extent that similar, but not identical, words are chosen in noninitial sentences. In general it may be said that the restriction pattern is

designed to effect an overall pattern, whereas the reweighting tables are more local in effect, dealing with purely lexical materials.

5.1. Development

An examination of hundreds of sentence pairs, and scores of paragraphs, of Russian scientific texts, suggests that the following principles of development are commonly employed in intersentence connection: (1) progress from the general to the specific (more rarely, the reverse); (2) from whole to part, or from multiplicity to singularity (presumably a variation of the first-cited principle); (3) past action to present; (4) "other" to "present" agent; (5) "other" to "present" place; (6) cause to effect (more rarely, the reverse); (7) action to purpose of the action; (8) action to means of performing the action; (9) simple rephrasing. Lack of space prevents illustration of these principles; it should be obvious that even this small stock of strategies will suffice for the production of innumerable paragraphs. It should also be noted that a random ordering of sentences built on the above pair-wise strategies will produce less than satisfactory results; certain sequences of sentence pairs are more likely than others to fit into an acceptable pattern for the paragraph.

It was stated in Sec. 1 that the computer program would provide a means of inspecting the initial sentence of a paragraph before deciding on a strategy for further develop-



Pattern 1

Pattern 2

Pattern 3

Pattern 4

Fig. 5 — Restriction patterns for a paragraph

a. The nature/ of scattering/ was investigated/ in an earlier paper.
NØ(+NØ) N(+N) VT(+C2) DV(+C19)

b. Belov/ proposed/ in paper (1)/ a means/ of determining/
NS(+C16) VT(+C7) DV(+C19) NØ(+NØ) N(+N)

the probability/ of absorption.

c. A theory/ of interaction/ is worked out/ in the present paper.
NO(+NO) N(+N) VT(+C7) DV(+C20)

d. A method/ of analyzing/ the method/ of analyzing/ the
NO(+method) N(+N)

magnitude/ of distortion/ is proposed.
VT(+VT)

The use of patterns to control development is summarized in Table 3. Since the verb investigate in sentence (a) belongs to semantic class C2, and C2 contains such verbs as "study" and "investigate" which specify very general actions, the node VT(+C7) in pattern 2 controls the selection of a verb of greater specificity; the verbs in C7 are appropriate. The node VT(+C7) in pattern 3 serves this purpose, i.e., to control the development of actions from generality to specificity. The node NS(+C16) in pattern 2 specifies that an agent for the second sentence is not the present author implicitly specified in the third sentence. This restriction introduces another type of text development, i.e. from other writer to present author. The node DV(+C19) in pattern 1 and pattern 2, and node DV(+C20) in pattern 3 introduce the time progression and location change to the paragraph. Class C19 contains such adverbs as "in an earlier paper," "in paper 1," "in an earlier study," etc., which specify that the time is past. Class C20 contains such adverbs as "in the present work," "in the present paper," etc. which specify the different locations in which some actions were performed.

Table 3

AN ILLUSTRATION OF THE USE OF RESTRICTION PATTERNS

Pattern	Elements	Coherence	Development
1	Action		
	Object		
	Agent: unspecified		
	Time: past		
	Location: earlier paper		
2	Action	Related to 1	Increase in specificity
	Object		
	Agent: specified as not the present author	Same class as 1 (researchers)	Nonidentical agent
	Time: past	Same as 1	
	Location: paper (1)	Same class as 1 (location)	Nonidentical location
3	Action	Related to 1, 2	Increase in specificity (relative to 1)
	Object		
	Agent: present author	Parallel to 1, 2	nonidentical agent
	Time: present		Time progression (past to present)
	Location: present paper	Same class as 1 (location)	Nonidentical location
4	Action	Related to 1, 2, 3	Increase in specificity
	Object	Restrict the selection of verb	
	Agent: present author	Same as 3: related to 1, 2	Same as in 3; different example for 1, 2
	Time: present	Same as 3	Different example for 1, 2; same as in 3

Table 3 also suggests partial introduction of coherence into the sentence-sequence. For example, the verbs in all sentences are in some degree related, and a general parallelism is maintained in the selection of agents and adverbs of time and location. Nonetheless it is clear that sentences a. through d. do not form a good paragraph. One deficiency is the excessively general character of the noun phrase, "the nature of scattering," in a.; a more obvious shortcoming is the lack of continuity in the noun object. Such deficiencies suggest the need for greater cohesion.

5.2. Cohesion

As a first approximation we have chosen to implement the following principles of cohesion: (1) selection of a "concrete" word in noun phrases; (2) word repetition; (3) use of hypernyms and synonyms; (4) use of anaphoric words; (5) increased repetition of members of the same semantic classes; (6) avoidance of word repetition within a single noun phrase. The implementation of these tactics is carried out in the reweighting table described in Sec. 4. In essence each tactic is a criterion for determining which words and semantic classes, together with reweighting values, are to be entered into the reweighting table. The content of the table depends on the criteria applied for each word selection; the main generation routine accepts the table as input for control of the selection, without "knowing" which criteria are used in forming the table.

When the principles of cohesion have been applied, sentences a. through d. above might have the following form:

a'. The nature of nuclear scattering was investigated in an earlier paper.

b'. Belov proposed in paper (1) a means of determining the probability of such pheonomena.

c'. A theory of proton scattering is worked out in the present paper.

d'. A method of analyzing the interaction of these particles is proposed.

The following are some of the improvements in this sentence-sequence over a. through d.:

(1) The addition of the "concrete" adjectives nuclear and proton gives the noun phrases in a'. and c'. a specificity that is lacking in a. and c. This effect is forced upon the generation routine by requiring the selection of dependents in a noun phrase to continue until a word coded as "concrete" has been chosen. (Since the effect may also be one of very long noun phrases, a countereffect is achieved by constant up-weighting of the semantic class of concrete words in the reweighting table.)

(2) The recurrence of scattering in a'. and c'. increases continuity in the sentence sequence. The generation program achieves word repetition in adjacent or nearly adjacent sentences by entering the noun-subjects or noun-objects of previously generated sentences (the

choice between noun-subject or noun-object is made by reference to the restriction pattern for the sentence being generated) into the reweighting table, together with a high positive reweighting value. Moreover, the possible governors of these nouns are also entered into the table with the same reweighting value. The value controls the probability of repeating one of the nouns in a previously generated sentence or of selecting a noun which is the governor of a word in a previously generated sentence. In the latter case word repetition will occur on the next level of dependency structure, i.e., when the program selects a dependent for the selected governor.

(3) The selection in b'. and d'. of phenomenon and particle, hypernyms of scattering and proton respectively, introduces semantic continuity and, in addition, reduces the redundancy and monotony of word repetition. The use of hypernyms and synonyms is implemented by entering any hypernym and synonym of the words in previous sentences into the reweighting table with a positive reweighting value, thus increasing the probability of their selection.

(4) The hypernyms phenomenon and particle in b'. and d'. acquire "concreteness" by the addition of anaphoric dependents such and these. The concreteness of the noun phrase such phenomena in b'. has presumably been provided by the dependents of scattering in a'. In the present system the addition of an anaphoric dependent for a hypernym

automatically terminates the selection of other dependents for the hypernym.

(5) The selection of proton and interaction in c'. and d'. is a result of increasing the repetition of members of the same semantic class: the semantic classes represented by nuclear in a'. and scattering in c'. are upweighted during the generation of c'. and d'.

(6) The undesirable repetition in d. is eliminated: words generated in a noun phrase are entered with negative value in the reweighting table, so that their repetition in the same phrase is inhibited.

The results of implementing even these few simple cohesion principles are encouraging. Experimentation with additional constraints continues.

6. CONCLUSION

The paragraph generator is currently operational, and produces output in reasonable times. Using the strategies for achieving development and cohesion so far developed, it is capable of generating ten-sentence strings in approximately fifteen seconds. Some of the main difficulties connected with the output are the following:

(1) Deficiencies in the co-occurrence data affect the quality of individual sentences. For example, some nouns have very few dependents, a characteristic deriving from their behavior in the text on which the data is based; the selection of one of these nouns in a sentence may nullify the effect of applying strategies for development or cohesion. In general a generated paragraph is only as strong as the weakest link; defective single sentences can disturb the implementation of structural principles.

(2) The grammar permits the generation of simple sentences only. Complex or compound sentences can, of course, be created by the device of juxtaposing these simple sentences with the help of conjunctions or relatives; the conditions under which this can be done remain to be specified.

(3) The creation of "lexical fields" (containing, e.g, such words as "to photograph," "camera," "film,") would greatly increase the effect of cohesion. Distributional data for the formation of such "fields" is not readily available; if the classes are to be intuitively created,

the result will be inconsistent with our present system of classification.

Study of these problems continues through analysis of the output. The effects of strengthening or relaxing various criteria for achieving development and cohesion have been observed in a series of experiments. The use of alternative sets of language input data (e.g., different dependent probabilities or semantic classes) is also contemplated. (It should be emphasized that the program is not oriented on a particular language or set of language data.) The experimental design of the generation program is consistent with this kind of modification.