Information Processing & Management, Vol. 11, pp. 39–57. Pergamon Press 1975. Printed in Great Britain

LSP-13

GRAMMATICALLY-BASED AUTOMATIC WORD CLASS FORMATION

LYNETTE HIRSCHMAN, RALPH GRISHMAN and NAOMI SAGER
Linguistic String Project, New York University, NY 10003, U.S.A.

(*Received* 1 *February* 1975)

Summary—Most previous attempts at producing word classes (thesauri) by statistical analysis have used very limited distributional information such as word co-occurrence in a document or a sentence. This paper describes an automatic procedure which uses the syntactic relations as the basis for grouping words into classes. It forms classes by grouping together nouns that occur as subject (or object) of the same verbs, and similarly by grouping together verbs occurring with the same subject or object. The program was applied to a small corpus of sentences in a subfield of pharmacology. This procedure yielded the word classes for the subfield, in good agreement with the word classes recognized by pharmacologists. The word classes can be used to describe the informational patterns that occur in texts of the subfield, to disambiguate parses of a sentence, and perhaps to improve the performance of current information retrieval systems.

BACKGROUND

SINCE the early days of computing, people have used statistical techniques to study the patterns of word usage in large bodies of text. These studies have been used in such diverse areas as stylistics, authorship determination, and information retrieval. Within information retrieval, one particular goal has been the automatic preparation of thesauri—lists of synonymous or related words—from word co-occurrence patterns in the texts [1, 2]. These thesauri can then be used to organize the data base and to enhance recall and precision.

A major limiting factor in such analyses has been the small amount of text structure utilized in the analysis. Most systems use only the most physically evident structure: for a collection, the division into individual documents; for a single text, the division into sections and occasionally into paragraphs or sentences. The grammatical relationship between words within a sentence is entirely lost. Such a system may determine that two words co-occur in a sentence, but cannot know whether they appear in a subject–verb relation, a host–modifier relation, or no relation at all. In order to recover this structural information, the sentences must be analyzed syntactically; because of the large volume of text usually involved, computerized syntactic analysis is essential.

Over the past decade, the Linguistic String Project has been developing a system for the automatic syntactic analysis of English scientific texts [3]. This system involves two stages of processing: sentence segmentation and transformational decomposition. The sentence segmentation component has been in operation for several years and is capable of segmenting the large majority of sentences in scientific texts. The transformational component has been under development for only a year; we anticipate that another one or two years will be required to prepare a set of transformations adequate for processing scientific texts. Because the string segmentation is designed to divide the sentence in a way which reflects its transformational composition, this task is proving to be relatively straightforward.

In parallel with this development effort, the Project has begun studying techniques for utilizing the wealth of information available in syntactically analyzed texts. In particular, we have been interested in the syntactic structures found in texts of specialized areas of science. An earlier study [4] indicated that the parts of the sentence carrying the scientific information fell into a small number of patterns, called information formats: certain groups of verbs occurred only with certain other groups of nouns as subjects and objects. Furthermore, these groups correlated closely with the intuitive semantic classes in the field. This suggested that word classes pertinent to the informational structure of the sentences could be obtained from an analysis of the subject–verb–object co-occurrence statistics.

To investigate this possibility further, we have syntactically analyzed by hand a number of texts, producing the same structures which will be generated automatically by our parsing system. These structures have been subjected to a computerized co-occurrence analysis which is described in detail in the rest of this paper. We have found that, by using this structural information, the co-occurrence analysis can uncover the classes of related words in particular science subfields.

Once these word classes are obtained, they can be used in a variety of ways. They can be used as a subfield thesaurus. The co-occurrence patterns of the word classes can be used to identify the informational structures of the sentences, i.e. to establish automatically the information formats for subfield sentences. The classes and their distribution patterns can also improve the syntactic processing of texts, by providing a means to distinguish between probable and improbable readings (parses) of a sentence which is syntactically ambiguous.

(1) OVERVIEW OF THE PROCEDURE FOR CLASS FORMATION

The clustering program groups words into classes on the basis of similarities in their distribution in the various texts analyzed. The co-occurrence of a certain noun with certain verbs but not with others reflects the informational role of the noun in the sublanguage* (and similarly for verbs). For example in the sublanguage under investigation (pharmacology articles on the cellular mechanism of digitalis action), we find phrases like:

(1.1) Potassium loss from the heart caused by CG (LE711 10.3.1)†

(1.2) ouabain did not interfere with the phosphorylation of the enzyme (LE711 11C.1.7)

In these sentences, and many others in the subfield texts, certain physiological processes are described in the kernel sentences‡: *the heart loses potassium*, *an enzyme is phosphorylated.* The drug words *CG* (cardiac glycosides) and *ouabain* are connected to the kernel sentence by two-place operators like *affect* or *interfere with*, non-kernel verbs that can also connect sentences. The pattern of syntactic occurrences reflects the information being conveyed: the drugs are introduced from the outside and are the active agents in these articles; drug action is described in terms of how the drug affects certain physiological systems: the heart, the cells, enzymes, etc., mentioned in the kernel level material. Because different kinds of nouns occur in different parts of the sentence, with different verbs, it is possible to use distribution to separate both the nouns and the verbs into sublanguage classes.

The input to the program consists of linearized tree representations of the sentences of a text. These representations are obtained manually by applying standard English transformations to the sentence.§ These transformations undo passives and nominalizations of verbs, expand conjunctional constructions, etc., as illustrated in section 2. The transformed sentence is represented by a tree made up exclusively of terminal nodes labelled with the base forms of the lexical items, arranged in an operator-operand hierarchy: e.g. the verb dominates its subject and complement(s), negation dominates the sentence (or noun phrase) that it negates. (1.1) is represented as follows:

(1.3) potassium loss from the heart caused by CG

```
        cause
        /   \
      CG    lose⟨vn⟩
            /   \
        heart   potassium
```

⟨vn⟩ stands for a nominalizing suffix (including zero) or a nominalizing vowel change.

A program then decomposes the tree into operator-argument pairs. For example the tree (1.3) yields the following pairs:

(1.4)	operator–first argument	operator–second argument
	(cause, CG)	(cause, lose)
	(lose, heart)	(lose, potassium).

These pairs serve as the input for the similarity coefficient computations on the lexical items.

Clusters are made up by grouping together "similar" lexical items. Two lexical items are similar if either the two words appear in a certain argument position under the same operator, e.g.

*We use the term sublanguage to refer to the specialized use of English in a particular subfield of science.

†The code *LE711 10.3.1* identifies a sentence in a text: *Lee* 197*1*, article *1*, section *10*, paragraph *3*, sentence *1*. A list of the pharmacology texts and their codes appears with the references.

‡A kernel sentence is defined here as a sentence with a verb that takes as its subject and object(s) only concrete nouns. For example *heart loses potassium* is a kernel sentence, but *digitalis causes potassium loss* is not, since the object of *cause* is *potassium loss*, a transformed sentence and not a concrete noun.

§Lists of standard English transformations can be found in[5–7].

both as subject of a certain verb; or both words operate on the same operand in a certain argument position, e.g. both have an occurrence with the same word as object. In addition both words must have the same argument structure, that is, take the same number and type of arguments. Concrete nouns take no arguments; for operators which take arguments, they take one of two types of argument: a concrete noun (N), or something which is itself an operator (S).

A similarity coefficient (SC) is computed for all possible pairs of words. Two words are clustered if their SC (based on the frequency of occurrence with the same operator or operand) exceeds a variable threshold value t. Clusters are built up one word at a time. A word is added to a cluster C_n to form a cluster C_{n+1} if for each word in C_{n+1}, the average of its SC with each other word exceeds the threshold. Clusters that are subsets of other clusters are not printed out. This method produces a number of clusters of varying sizes. Some clusters overlap partially and are merged to form a single larger class, provided that the overlap is sufficiently large. A cluster is merged into another cluster if p% of the first cluster's members are also members of the second cluster. The *merged classes* are the word classes of the sublanguage. These word classes are presented in Table 2 below. Sections 2–5 describe each step of the process in greater detail.

(2) GENERATION OF OPERATOR-OPERAND PAIRS

(a) *Trees*

Each sentence is represented as a tree with only lexical items as node labels, with each operator (verb) node dominating its argument (subject and object) nodes.* In order to represent a sentence in tree form, it must first undergo transformational decomposition into subject–verb–object units. In this study the trees were made manually, using transformations which are currently being added to the computer processor. We attempted to simulate the computer transformational analysis as closely as possible; however, in cases where more than one analysis was syntactically correct, we chose the intended reading for further processing.†

The transformations used in decomposition preserve the informational content of the sentence, but regularize the co-occurrence patterns (for example by changing passive to active, so that all forms are in the active; or by changing a complex noun phrase containing a nominalized verb with prepositional phrase to a subject–verb–object pattern). For example:

(2.1) Ca^{++} uptake of SR

(2.2) SR takes up Ca^{++}

Word sequences (2.1) and (2.2) clearly carry the same information; the nominalization transformation (2.3) can be used to reduce (2.1) to the subject–verb–object form in (2.2):

(2.3) N_1 nom(V) of $N_2 \leftrightarrow N_2\ V\ N_1$.

nom (V) stands for the nominalization of a verb, e.g. "*uptake*" from "*take up*".

In the trees, words are reduced to a standard form (i.e. *uptake* is changed to its infinitive form *take up*). The original nominalizing suffix is noted in angle brackets after the word, as are prepositions from the nominalization or prepositional objects.

The tree from 2.1 or 2.2 is:

(2.4)
take up ⟨vn⟩
SR / \ Ca^{++}

Similarly 2.5 is related to the subject–verb–object form 2.6:

(2.5) the exchange of Ca^{++} with cations

(2.6) Ca^{++} exchanges with cations

by the following transformation:

(2.7) nom(V) of N_1 prep $N_2 \leftrightarrow N_1\ V$ prep N_2

The tree below represents 2.5 and 2.6‡:

exchange ⟨vn⟩ ⟨with⟩
(2.8) Ca^{++} / \ cation.

*This type of structure corresponds to the operator-operand formalism of HARRIS[8]. It has also been called a dependency tree[9].

†Alternatively, it may be possible to include all parses in the statistical analysis, with the correct grammatical pairings (which will occur repeatedly) dominating the incorrect pairings (which should be randomly distributed) over a large body of text.

‡Nouns are reduced to their singular form.

Kernel sentences 2.1 and 2.5 are both found embedded in a more complex sentence:

(2.9) Carvalho and Leo found that the Ca^{++} uptake of skeletal SR involves the exchange of Ca^{++} with other cations in SR. (LE711 13C.5.7)

In sentence 2.9 some of the nouns appear with adjuncts (modifiers), e.g. *cations* occurs with *other* and *in SR*. When an adjunct includes a sublanguage noun (*in SR*), this material is expanded into a relative clause adjunct. Other adjuncts (e.g. *other* on *cation*, and *skeletal* on *SR**) are represented to the side of the noun, connected by a single dash:

(2.10)

```
     exchange⟨vn⟩⟨with⟩
          /    \
      Ca++   cation - other
```

Relative clauses and derived relative clause constructions are handled as follows: relative clauses are found attached to a "head" noun phrase:

(2.11) $\underbrace{\text{cations}}_{\text{head}}\ \underbrace{\text{which are in SR}}_{\text{relative clause.}}$

In the relative clause, the relative pronoun appears in place of the head noun phrase (*which* in place of *cations*). In order to obtain the usual subject–verb–object relations from the relative clause, we replace the relative pronoun by the head noun. The "filled out" relative clause sentence then hangs from the relative pronound, which is attached to the side of the head noun like an adjunct, except with two dashes:

(2.12)

```
          exchange ⟨vn⟩ ⟨with⟩
           /            \
      Ca++        other - cation - - [wh]
                                      |
                                      in
                                     /  \
                              [cation]   SR
```

The repeated head noun *cation* appears in square brackets in the relative clause; square brackets [] are used to enclose all uniquely recoverable implicit material (said to be "zeroed"). In the phrase *cations in SR* (sentence 2.9), the relative pronoun has itself been zeroed; therefore a *wh* is reconstructed as the relative pronoun and, like *cation*, enclosed in brackets. *In* is taken as the operator in the relative clause. We could have taken *be in* as the operator, but since *be* serves merely as a carrier of tense, it is omitted with prepositions and adjectives, even when it does occur explicitly.

We can now draw the entire tree representation for sentence LE711 13C.5.7:

(2.13)

```
                         find ⟨that⟩
                 /                     \
             and                       involve
            /   \                   /           \
     Leo   Carvalho        take up ⟨vn⟩       exchange⟨vn⟩⟨with⟩
                            /      \            /          \
                   skeletal - SR   Ca++      Ca++   other - cation - - [wh]
                                                                        |
                                                                        in
                                                                       /  \
                                                                [cation]   SR
```

(The number and order of adjuncts on a noun, e.g. those on *cation* above, is immaterial for the clustering program.) *And* appears as a two-place connective, with *Leo* and *Carvalho* as its arguments. However, the pair-generating program, explained in 2B, treats *and* as "transparent": that is, when it looks for the first argument of *find* it looks through *and* to the arguments of *and*, and forms two operator-1st argument pairs: (*find*, *Leo*) and (*find*, *Carvalho*).

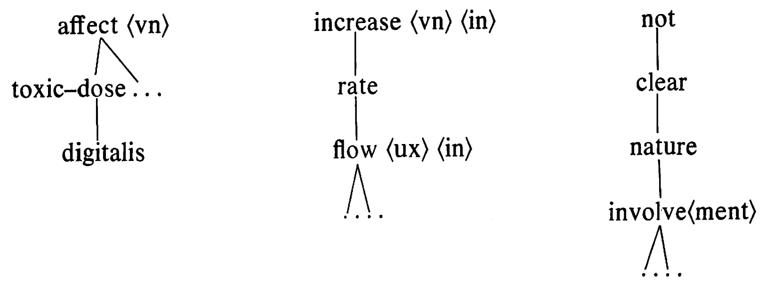
**Skeletal* is not decomposed into *skeleton* because there are no occurrences of *skeleton* in the sublanguage texts.

One final point should be explained: there are a number of constructions N_1 *of* NP_2 (NP = noun phrase), where N_1 is not derived from a verb, but is also clearly not what is ordinarily considered a concrete noun:

(2.14) *effect of a toxic dose of digitalis*; *increase in the rate of influx*; *the nature of the involvement is not clear.*

In these cases, it was decided to treat these nouns (e.g. *does, rate, nature*) as one-place operators on the NP_2:

(2.15)



This is not an intuitive representation of this construction, and we are still searching for a more linguistically satisfying treatment. (This problem arises because the general English transformations for dimension words are very poorly understood.)

(b) *Generation of pairs*

Each tree is linearized and processed to yield operator–argument pairs. To linearize a tree (illustrated in 2.16 below), each word, together with its arguments, if any, and modifiers (adjuncts and relative clauses) is enclosed in parentheses. The linearized tree is then decomposed to yield three distinct types of pairs: operator–first argument, operator–second argument and operator–third argument. Since almost no verbs take more than three arguments, no allowance is made for more than three arguments. All material in angle brackets (suffixes, prepositions) is ignored in making up the pairs; most of this material reflects the transformations that a word has undergone to reach its base form (the form with no suffixes). In general this information is not relevant to the clustering; in fact it would not be desirable to treat as separate words two forms of a single word, e.g. *augment⟨tion⟩* vs *augment⟨ed⟩*. If a case should arise where this information is needed however, it is still available in the tree. Material in brackets [] is treated just as unbracketed material. Adjuncts are ignored, although host–adjunct pairs could be produced if desired.

(2.16) Linearization of example 2.13:

(find ⟨that⟩
  (and (Leo) (Carvalho))
  (involve
    (take up ⟨vn⟩
      (SR-(skeletal)) (Ca^{++}))
    (exchange ⟨vn⟩ ⟨with⟩
      (Ca^{++}) (cation-(other) - - ([wh](in([cation])(SR)))))).

(2.17) Pairs generated from 2.13:

operator–1st argument	operator–2nd argument	operator–3rd argument
(find, Leo)	(find, involve)	
(find, Carvalho)	(involve, exchange)	
(involve, take up)	(take up, Ca^{++})	
(take up, SR)	(exchange, cation)	
(exchange, Ca^{++})		

Note that no pairs are generated with *in* as operator. This is because *in* is a structural operator, and is therefore ignored, as explained in (4) just below.

A number of grammatical word classes are treated in a special manner (these classes are listed in Appendix 1):

(1) Binary connectives (*and, but, or,* etc.) are transparent: the tree processor looks through a transparent word without incorporating it into a pair, and takes as the arguments of the operator on the transparent word the arguments of the transparent word (see treatment of *and* in the example above).

(2) Modals (*can, will,* etc.), aspectuals (e.g. *seems to*), and negatives are also transparent, but take only one operand.

(3) Relative clause connectives (*wh, which,* etc.) are "ignored", i.e. no pairs are formed with them. They are not in fact operators at all, but are used to mark the head noun of the relative clause.

(4) Structural operators (e.g. *have, in, constitute*), *be* and *be*-like verbs are ignored. The word-class program groups words together that appear as subject (or object) of a given operator. With structural and *be* verbs, however, there is no similarity between all first arguments or all second arguments.

Example:

(2.18a) *ATPase is an enzyme*
(b) *digitalis is a drug*
} *ATPase* and *digitalis* are not similar, nor are *enzyme* and *drug.*

The important relation here is between the operands of the same operator, not between the operand and operator. Therefore these words are not clustered in the usual manner. It remains to work out a way to use this information in the formation of word classes.

(5) Subordinate conjunctions (*since, if,* etc.) are ignored, for reasons similar to those for ignoring the structural and *be* operators: two first arguments of *if* may have nothing in common, since *if* can be used to connect almost any two sentences in English.

(6) Verbs which occur in a middle voice construction (e.g. *increase, diminish*) and which also occur in a causative construction are treated in a special way. We can find: *the concentration increased, digitalis increased the concentration, digitalis increased the influx*; in one instance the first argument of *increase* is *concentrate,* and in the others it is *digitalis,* with *concentrate* or *influx* appearing in second argument position. Clearly this does not give the desired kind of alignment: *concentrate* and *influx* are parallel, and not *concentrate* and *digitalis.* To remedy this, if these verbs occur with only one argument (i.e. in middle voice construction), the program takes the single argument to be the second argument.

(3) COMPUTATION OF THE SIMILARITY COEFFICIENT

Each word W_i is assigned a characteristic vector V_i on the basis of its co-occurrence in particular grammatical relations with other words in the text. If there are n distinct words in the corpus, the characteristic vector for any word will have $6n$ components, because each word W_i can appear in any one of six possible relations to a given word W_j:

(1) W_j is an operator and W_i is its first argument
(2) W_j is an operator and W_i is its second argument
(3) W_j is an operator and W_i is its third argument
(4) W_i is an operator and W_j is its first argument
(5) W_i is an operator and W_j is its second argument
(6) W_i is an operator and W_j is its third argument

Since exceedingly few operators take four arguments (a subject and three objects), this fourth argument position (third object) has been ignored in the calculations. The value of the component indicates the number of pairs in which W_i and W_j appear in that particular relation. All the characteristic vectors are sparse: only a few of the several thousand components are non-zero.

Each vector is divided by a normalization factor to produce a vector of unit length. (The normalization factor for a vector is the square root of the sum of the squares of its components.) The vector may also be multiplied by a weighting factor, discussed below. The similarity coefficient between two words W_i and W_j is the inner product of the normalized, weighted characteristic vectors of the two words:

$$SC_{ij} = V_i \cdot V_j = \sum_{k=1}^{6n} (V_i)_k \cdot (V_j)_k$$

Example:

(3.1) Similarity coefficient for *depress* and *alter* (data from 11.13.74, shown in Table 1). The table lists only the non-null component vectors for *depress* and *alter*; the entry in the table is the number of times W_i occurs in that pair.

Table 1

(operator,	operand)	W_i depress	W_i alter	
(1) W_i,	Σ magnesium	2	0	
(2) W_i,	Σ oligomycin	1	0	(W_i, Σ CG) = *CG* as subject of W_i
(3) W_i,	Σ quinidine	1	0	
(4) W_i,	Σ acetylstrophanthidin	0	1	
(5) W_i,	Σ *CG*	0	1	
(6) W_i,	Σ digitalis	0	2	(shown, Ω W_i) = W_i as object of *show*
(7) W_i,	Σ drug	0	1	
(8) W_i,	Σ present	0	2	
(9) W_i,	Ω act	1	2	
(10) W_i,	Ω contract	1	0	
(11) W_i,	Ω enzyme	1	0	
(12) W_i,	Ω transport	3	1	
(13) W_i,	Ω bind	0	1	
(14) W_i,	Ω distribute	0	2	
(15) W_i,	Ω exchange	0	2	
(16) W_i,	Ω property	0	1	
(17) W_i,	Ω structure	0	1	
(18) W_i,	Ω take up	0	1	
(19) W_i,	Ω utilize	0	1	
(20) in such way,	Σ W_i	0	1	
(21) possible,	Σ W_i	0	1	
(22) secondary,	Σ W_i	0	1	
(23) assoc. with,	Σ W_i	1	0	
(24) like,	Σ W_i	1	0	
(25) show,	Ω W_i	0	1	
(26) report,	Ω W_i	2	0	

Normalization factor $\begin{cases} \textit{depress} = \sqrt{(24)} \\ \textit{alter} = \sqrt{(33)} \end{cases}$

Similarity coefficient: $((1 \times 2) + (3 \times 1))/(\sqrt{(24)} \times \sqrt{(33)}) = 5/\sqrt{(792)} = 0{\cdot}178$; only lines 9 and 12 contribute.

The weighting factor is introduced to deal with low frequency words. For example, in the data of 11.13.74 there is only one occurrence of *small bowel* (as the object of the verb *affect*), and only one occurrence of *membrane ATPase* (also as the object of *affect*). As a result these two words have a similarity coefficient of 1·00, based on a single occurrence with a very general verb, *affect*. To avoid the formation of such false clusters of low frequency words, the normalized vector for each word W_i is multiplied by a weighting factor which gives less weight to infrequently occurring words:

Weighting factor for word $W_i = 1 - (0{\cdot}99/\sqrt{(n)})$ where n = the number of occurrences of W_i in operator–operand pairs. This weighting factor virtually eliminates clustering on the basis of a singly occurring word: if W_i occurs only once, then its weighting factor multiplies the characteristic vector by $1 - (0{\cdot}99/\sqrt{(1)}) = 0{\cdot}01$.

Example:

(3.2) Weighted similarity coefficient for *depress* and *alter*:

Weighting factor for *depress* (n = 14): $1 - (0{\cdot}99/\sqrt{(14)}) = 0{\cdot}735$

Weighting factor for *alter* (n = 23): $1 - (0{\cdot}99/\sqrt{(23)}) = 0{\cdot}793$

weighted $SC_{\text{depress-alter}} = (5)\ (0{\cdot}735)\ (0{\cdot}793)/(\sqrt{(24)} \times \sqrt{(33)}) = 0{\cdot}103$.

(4) CLUSTERING PROCEDURE

Two words form a cluster if their similarity coefficient (calculated as described in the previous section) exceeds the threshold t. Clusters are built up one word at a time. This avoids the problem of grouping two unrelated subclasses of words together (illustrated in example 4.3).

A word may be added to a cluster C_n to form a new cluster C_{n+1} if and only if, for each word in

C_{n+1}, the average of its SCs with each other word in C_{n+1} is greater than or equal to the threshold.

(4.1) Given words $W_1, \ldots, W_n$ which form a cluster C_n. Then $C_n \, U\{W_{n+1}\}$ is a cluster iff:

$$\underset{1 \leq i \leq n+1}{V_i} \left[\left(\frac{\sum_{1 \leq j \leq n+1\, j \neq i} (SC_{ij})_{j \neq i}}{n} \right) > t \right]$$

Example:

(4.2) $t = 0{\cdot}3$, SCs from data of 7.9.74

SC of *digitalis* with *CG*: 0·536; *Digitalis* and *CG* form a cluster.

Can *drug* be added to this cluster, to get a three-word cluster?

$SC_{drug/digitalis} = 0{\cdot}316$; $SC_{drug/CG} = 0{\cdot}386$

$(SC_{drug}) = (0{\cdot}386 + 0{\cdot}316)/2 = 0{\cdot}351$ (greater than t)

$(SC_{digitalis}) = (0{\cdot}536 + 0{\cdot}316)/2 = 0{\cdot}426$ (greater than t)

$(SC_{CG}) = (0{\cdot}386 + 0{\cdot}536)/2 = 0{\cdot}461$ (greater than t).

Since all the averaged similarity coefficients exceed the threshold, *drug* can be added to form a three-word cluster: {*drug, digitalis, CG*}. Finally, can *ouabain* be added to this, to form a four-word cluster?

	CG	drug	digitalis	ouabain	average
CG	×	0·386	0·536	0·476	0·499
drug	0·386	×	0·316	0·248	0·317
digitalis	0·536	0·316	×	0·171	0·341
ouabain	0·476	0·248	0·171	×	0·298

The average of the *ouabain* SCs does not exceed the threshold value of 0·3, hence it cannot be added to form a large cluster. However, if the threshold were lowered, say to 0·25, then it could be added to form a four-word cluster. Note also that the three words *CG, drug, ouabain* form a cluster at $t = 0{\cdot}3$, as do *CG, digitalis, ouabain.* This example indicates why there are often a number of similar clusters, with almost identical members. A slight lowering of the threshold may allow these words to combine into one large cluster. (The merging procedure also collapses these two clusters into one.)

If the words were not added to the cluster one at a time, then a cluster might be formed from two unrelated sets of words, as illustrated in the following example:

Example:

(4.3) SCs from data of 7.9.74.

Taking the threshold to be 0·2, the following four words meet the criterion of the average of similarity coefficients, but the cluster cannot be formed by adding one word at a time:

	CG	digitalis	Na + K + ATPase	ATPase	average
CG	×	0·536	0·072	0·069	0·226
digitalis	0·536	×	0·137	0·143	0·272
Na + K + ATPase	0·072	0·137	×	0·655	0·288
ATPase	0·069	0·143	0·655	×	0·289

The very high SC of the pairs of related words in example 4.3 is enough to compensate for the low SC between the less related words (drug/ATPase). However, since neither of the drug words (*CG, digitalis*) is related to the ATPase words, no intermediate three word cluster can be formed, hence the set {*CG, digitalis, Na + K + ATPase, ATPase*} is not a cluster.

(5) MERGING PROCEDURE

Appendix 2 lists the clusters generated by the procedure described in sections 2 through 4. It is difficult to interpret this set of clusters because of the overlap between classes. For example, there are 13 clusters of *CG* words. The question arises: are these distinct subclasses, or is this an artifact of the clustering algorithm? The merging procedure is based on the assumption that if there are two largely overlapping classes, then these are actually subsets of a single larger class.

The merging procedure works as follows: A cluster is treated as a nucleus around which words can collect. Each class is checked against this nucleus and if a class resembles the nucleus, its members are included in the *merged class* (the class formed around the nucleus). A class *resembles* the nucleus if $p\%$ or more of its members also belong to the nucleus. Every class in turn is treated as a nucleus which produces a new merged class. If no other class resembles the nucleus, the merged class will be identical to the nucleus; two distinct nuclei may produce two merged classes which are identical except for the ordering of the members. In this case, since the order of the members is immaterial, the set is not written down again. Also a merged class derived from one nucleus may be a subset of a merged class derived from another nucleus; in this case only the larger class is retained.* The procedure of merging classes is then repeated on the new set of merged classes until no new merged classes are obtained as a result of applying the merging procedure.

Example:

(5.1)

Suppose we have the following 3 clusters:

(a)	(b)	(c)
sodium	sodium	sodium
calcium	potassium	potassium
Ca	calcium	K
Ca^{++}		

Then using (a), (b) and (c) successively as nuclei, we get merged classes (MC): ($p = 66\%$)

(MCa)	(MCb) = (MCc)
sodium	sodium
calcium	potassium
Ca	calcium
Ca^{++}	K
potassium	

Applying the merging procedure a second time we get a single class:

(MC′a)		(MC′b) = (MC′a)
	sodium	
	calcium	
	Ca	
	Ca^{++}	
	potassium	
	K	

(6) WORD CLASS FORMATION: RESULTS

The clustering program was run on a set of 400 sentences taken from six texts on the mechanism of action of digitalis (see References). Sentences were not specially selected, except that the Methods section was excluded. Each sentence was decomposed using standard English transformations and represented as a tree structure (as described in section 2). The tree was processed to produce operator–argument (e.g. verb–subject or verb–object) pairs. The set of 400 sentences yielded approx. 4000 pairs and a vocabulary of some 750 words. The similarity coefficients between each pair of words was computed (as described in section 3); the similarity coefficients were then used to group the words into clusters (section 4; Appendix 2 for the list of clusters) and finally the clusters were combined into merged classes, by the merging procedure described in section 5.

The effectiveness of the word-class program can be evaluated on the basis of three criteria:

(1) Does each merged class produced by the program form a legitimate sublanguage word class; i.e. does the merged class include words that belong together and exclude words that do not belong to that class?

*This is consistent with the fact that a regular cluster is not printed out if it is a subset of a larger cluster.

(2) What proportion of the words belonging to a given class is captured in the grouping generated by the program?

(3) Are all relevant word classes obtained in this way, and if not, which classes are lost? For the material on digitalis, the set of classes generated by the program can also be compared to the subfield classes established on the basis of a larger corpus of data[10], summarized in[4]. While 400 sentences is a small corpus, it turned out, rather surprisingly, that the main subfield word classes and the main members in each class were obtained by the computer program. Table 2 displays the final output of the program (the merged classes) for the 400-sentence corpus. In addition, some high frequency words were not part of any cluster; these are considered single member classes. A word was considered to be of high frequency if it occurred in more than 25 pairs.

Some of the merged noun classes displayed in Table 2 are evaluated in Tables 3 and 4 by comparing them to the classes obtained manually for the same corpus. The manual classes are essentially semantic classes, prepared in consultation with a pharmacologist.

Tables 3 and 4 illustrate that the word classes produced by computer are indeed valid word classes, that they include the major nouns of each class, and with minor exceptions do not include nouns from other classes. The word classes shown in Tables 3 and 4 accounted for over 80% of

Table 2. Merged classes, Run of 11.13.74, $t = 0{\cdot}250$, $p = 0{\cdot}066$

NOUN CLASSES:

CG CLASS
agent
cardiotonic glycoside
CG
compound
digitalis
drug
erythrophleum alkaloid
inhibitor
ouabain
strophanthidin
strophanthidin 3 bromoacetate
strophanthin

CATION CLASS		
Ca	ion	ion
Ca^{++}	K^{+}	substance
calcium		
electrolyte		
glucose		
ion		
K		
Na^{+}		
potassium		
sodium		

MUSCLE CLASS
atrium
heart muscle
muscle
ventricle

PROTEIN CLASS
actomyosin
cardiac
fiber
protein

ENZYME CLASS
Na + K + ATPase
ATPase
enzyme

SR CLASS
sarcoplasmic reticulum
SR

FALSE CLUSTERS	
Myocardium	ADP
cell	El

VERB CLASSES: KERNEL LEVEL (words which operate on concrete nouns)

MOVE CLASS = $_{I}V_{C}$	
move	distribute
turnover	intra
extra	
intra	
concentrate	
flow	

EXCITE CLASS = $\_V_{M}$
excite
depolarize

LOSE CLASS = $_{C}V_{I}$
lose
contain

SLIDE CLASS = $_{P}V$
slide
fold

SPACE CLASS = $_{I}V$
space
milieu

Table 2 (*Contd*)

FALSE CLUSTERS

take	potential	transport
treat	species	exchange

VERB CLASSES: V_Q (verbs which operate on quantitative operators)

CHANGE CLASS	AUGMENT CLASS	FALSE CLUSTERS	
increase	augment	measure	trigger
change	improve	decrease	augment
decrease	increase		

VERB CLASSES: V_{SS} and V_{NS} (non-kernel relational verbs)

stimulate	relate	dissociate	correlate	diverge
inhibit	similar	relate	relate	similar
influence	link			oppose
reverse	due to			
reduce	demonstrate			
act	cause			
affect	produce			
induce				
effect				
cause				
produce				
interfere				
alter				
*concentrate				
*penetrate				
*toxic				

REPORT CLASS = $_{N_h}V_S$	FALSE CLUSTER
report	depress
observe	mechanism

*These three words are kernel operators but appear here because they occur frequently with *CG* words as subject. Since the similarity coefficient is presently computed on the basis of sharing one argument, words can be clustered together even if they do not share any second position arguments. Unless we require that two words share both subject and object for a non-zero similarity coefficient, this will remain a problem.

KEY TO VERB CLASS NAMES:

C = cell	M = membrane	$_XV_Y$ is a verb class whose first argument (subject) is X and whose second argument (object) is Y.
I = ion	— = unknown	
P = protein	$\bar{N}_h$ = human noun	

Note: parallel lists under a heading are unmerged classes which belong together. Since merging requires a 66% overlap, two-word clusters could not be merged into a larger cluster.

the pair-occurrences of words in that class. It now remains to answer question 3: are all relevant word classes obtained in this way, and if not, which classes are lost? This information is summarized in Table 5.

Of the 11 major noun classes found manually, 10 are accounted for by the computer: six by merged clusters and four by single member classes. One major class recognized manually (*phosphorylated compounds*) did not appear, due to a minor mistake in the program. On the average the computer classes accounted for 84% of the nouns in each manual class. Overall the computer classes + single member classes account for 1335 of 2016 occurrences = 66% of pair-occurrences of concrete nouns in the corpus.

The number of nouns incorrectly classified was low: seven nouns were inserted incorrectly into classes (out of 43 nouns classified). The number of their occurrences was less than 9% of the total occurrences correctly classified. In short, the word class program accurately generated the major noun classes of the sublanguage.

In the corpus analyzed there were almost twice as many verbs (operators) as concrete nouns (500 to 270 nouns). Most of the computer verb classes are small, however, because only verbs of the same argument type are clustered together. There are verb clusters of the types: 1-place, 2-place and 3-place kernel operators, non-kernel V_Q operators (on quantity words), and other non-kernel operators. The computer and manual classes of kernel operators are compared in

Table 3

CG CLASS COMPUTER	MANUAL	No. OCCURRENCES IN PAIRS†	
CG	CG	156	
digitalis	dititalis	118	
ouabain	ouabain	70	
drug	drug	15	
agent	agent	8	
strophanthidin	strophanthidin	5	$\frac{395}{442} = 89\%$
strophanthidin 3 bromoacetate	strophanthidin 3 bromoacetate	4	
strophanthin	strophanthin	4	
cardiotonic glycoside	cardiotonic glycoside	3	
compound	compound	7	
inhibitor	inhibitor	5	
*erythrophleum alkaloid		*6	
	glycoside	11	
	digoxin	7	
	acetyl strophanthidin	7	
	cardioactive glycoside	6	
	digitalis glycoside	6	
	digitoxigenin	3	
	sprophanthoside	2	
	cardiac glycoside	2	
	digitoxin	1	
	digitalis compound	1	
	strophanthin K	1	
		442	

**Erythrophleum alkaloid* does not belong in the CG class; it is a drug whose effect is compared to that of the cardiac glycosides.

Agent, drug and *compound* are classifiers for words of the CG class, as well as of the more general DRUG class. *Inhibitor* is also a classifier, which classifies according to function.

†An occurrence of a word either as the operator or operand in a pair. Pair-occurrences are more numerous than text occurrences for several reasons. Recoverably zeroed material is reconstructed and contributes to pair formation. Also each operator can appear in a pair as the operand of its operator, as well as with each one of its arguments. (Thus a two-argument verb can appear in three pairs.) For concrete nouns however this does not occur, and the pair-occurrences correlate more closely with the number of actual occurrences in the text.

Table 6. The manual classes each has a corresponding computer class, most of them single member classes.

Table 7 compares the manual and computer V_Q classes. No manual classes of the other non-kernel operators (V_{SS} and V_{NS} in Table 2) were established for comparison with the computer output for these types. The output in Table 2 indicates that there may be an interesting substructure to these (roughly, causal) relational verb classes.

Table 4

CATION CLASS COMPUTER	MANUAL	No. OCCURRENCES IN PAIRS	
calcium	calcium	101	
Ca^{++}	Ca^{++}	48	
Ca	Ca	30	
potassium	potassium	90	
K	K	29	$394/412 = 96\%$
sodium	sodium	53	
Na^{+}	Na^{+}	11	
ion	ion	15	
electrolyte	electrolyte	17	

Table 4 (*Contd*)

CATION CLASS COMPUTER	MANUAL	No. OCCURRENCES IN PAIRS
*glucose		*7
	K^+	6
	Na	3
	Magnesium	3
	Mg^{++}	3
	Cation	3
		412

**glucose* appears in the computer CATION class due to its occurrence as the object of *transport*, a central verb for the CATION class. Since glucose and the cations behave differently in other respects, one would expect them not to be clustered together if a larger corpus of sentences were used.

K^+ and *ion* are clustered together in a two-word cluster; presumably with a larger corpus, K^+ would be included in the larger cluster.

Table 5. A comparison of noun classes obtained manually and by computer

CLASS	MANUAL*	COMPUTER*‡	%COMP/MAN
	No. OCC/No. N	No. OCC/No. N	
major classes†:			
CG	442/ 22	395/ 11	89
CATION	412/ 14	394/ 9	96
ENZYME	192/ 13	157/ 3	82
PROTEIN	136/ 21	63/ 3	45
SR	101/ 5	97/ 2	97
CELL	82/ 6	77/ 1	94
PHOSPH. CMPDS.§	66/ 10	xxxxxxx	xx
MEMBRANE	55/ 5	42/ 1	76
HEART	53/ 3	39/ 1	74
HEART PARTS	44/ 3	35/ 1	80
MUSCLE	45/ 6	38/ 2	84
minor classes†:			
HUMAN AGENT	95/ 54		
DRUG, NOT INCL. CG	88/ 25		
ULTRASTRUCTURE, NOT INCL. SR	42/ 15		
NATIVE ORG. SUB.	33/ 12		
ORGANISM	23/ 9		
TISSUE	20/ 3		
ORGAN NOT HEART	17/ 8		
INORG. MOLECULE NOT INCL. CATION	15/ 6		
EXPT. MEDIUM	13/ 3		
PHYSICAL FORCES	12/ 5		
MISCELLANEOUS	52/ 12		

*Entries are: total number of pair occurrences of nouns in class/number of nouns in class.

†Major classes are classes which have 50 or more total occurrences, and at least one member with more than eight occurrences. Minor classes have either less than 50 occurrences total, or no member with more than eight occurrences, as in the human agent class.

‡Single member classes are shown in correspondence to manual classes if the single word in question accounts for two-thirds or more of the pair occurrences of words in the manual class. In almost all cases, this word is identical to the name of the class.

§A *phosphorylated compounds* class was obtained on previous runs (five nouns, with 71% coverage of the manual class). Due to a small error, this class did not appear in this run.

Table 6. Summary of manual and computer verb classes: Kernel verbs*

CLASS	MANUAL	COMPUTER	% COMP/MAN
	No. OCC/No. N	No. OCC/No. N	
1-place kernel verbs			
CONTRACT	183/ 2	178/ 1	97
FUNCTION	21/ 3	14/ 1	67
FAIL	23/ 1	23/ 1	100
SLIDE	12/ 3	8/ 2	67
RELAX	13/ 2	11/ 1	85
2-place kernel verbs			
MOVE	508/ 13	418/ 6	85
LOSE	336/ 7	217/ 2	65
INTERACT	129/ 6	xxxxxxx	xx
CONVERT	62/ 3	44/ 1	71
ACTIVATE	41/ 3	32/ 1	78
EXCITE	70/ 4	62/ 2	89
OXIDIZE	17/ 2	14/ 1	82
3-place kernel verbs			
CARRY	210/ 4	158/ 1	75
BIND	132/ 1	132/ 1	100
EXCHANGE	51/ 2	46/ 1	90
PHOSPHORYLATE	50/ 2	34/ 1	68
misclassifications			
take-treat	136/ 2		
exchange-transport	204/ 2		
potential-species	32/ 2		

**Be*-like and structural verbs are not clustered, as was noted in Section 2. Also experimental verbs (e.g., *sectioned*) and "part" operators (*part*, *group*, etc.) have not been listed here. Experimental verbs cover a wide range of laboratory techniques used on a number of different systems, with different reagents. Therefore it is not surprising that they were not recognized as a class by the computer.

Table 7. A comparison of the manual and computer V_Q classes

MANUAL	COMPUTER		No. PAIR-OCCURRENCES	
change	change		115	
decrease	decrease	CHANGE CLASS	71	323/516 = 63%
increase	increase	———	137	
augment	augment	AUGMENT CLASS	39	
improve	improve	———	12	374/516 = 73%
reduce			73	
alter			21	
depress			13	
develop			8	
lower			5	
prolong			6	
accumulate			3	
decay			2	
accelerate			2	
diminish			3	
elevate			2	
maintain			2	
hold constant			1	
keep constant			1	
slow			1	
			516	

NOTE: There are two computer V_Q classes generated: *change, decrease,* and *increase*; and *augment, increase, improve.*

(7) APPLICATIONS

One possible application of the clusters lies in improving recall and precision in current information retrieval systems. Experience with thesauri in information retrieval indicates that the possible benefit depends very sensitively on the nature of the clusters and how they are used [2]. It is therefore very difficult to predict the value of our clusters within the context of current, keyword-oriented, retrieval systems.

Another potential application for the clusters is the cataloging of the principal low-level constructions used in the sublanguage. This process, called syntactic formatting, has been described in other papers [4, 11]. A few, very preliminary, efforts have been made at automating this process using the output of the clustering process.

Suppose we call each node in a parse tree, together with its immediate descendants (i.e. a verb with its subject and possible objects), a *pattern*. A frequency analysis of the patterns themselves will not be very fruitful, since most will occur only a few times. If, however, each word is replaced by a name assigned to the cluster containing the word, the number of frequently occurring patterns should increase greatly. In fact, our manual efforts at formatting indicate that most lower level structures will fit one of a small number of such patterns. Patterns of a similar type have been identified in medical records [12].

If our manual efforts can be successfully automated in this way, we should be able to produce, from texts in a science subfield, a set of formats suitable for structuring the information in those texts. This should simplify considerably any further processing of the data in the texts.

The formats would also return dividends to the parsing process. The observation that certain classes of verbs can appear only with certain classes of operands can be formulated as a set of *sublanguage restrictions* and used to augment the general English grammar. This should greatly reduce the number of extraneous parses. For example, in LE711 11D.1.2,

...the stimulatory effect of CG on NA + K + ATPase in a low concentration range...

a purely syntactic analysis could not determine whether *in a low concentration range* modifies *ATPase* or *CG*. However, in the sublanguage of our corpus, *concentrate* takes as its first argument only members of the ION, CG, or DRUG classes, and does not appear with *ATPase* as its argument. This information can be used by the parsing program to select the intended reading.

Acknowledgement—This work was supported in part by research grants RO1 LM00720 from the National Institutes of Health, DHEW, and by GN39879 from the National Science Foundation, Office of Science Information Services.

REFERENCES

[1] G. SALTON: Experiments in automatic thesaurus construction for information retrieval. *Proceedings IFIP* (1971).
[2] K. SPARCK-JONES: *Automatic Keyword Classification for Information Retrieval.* Archon (1971).
[3] R. GRISHMAN, N. SAGER, C. RAZE and B. BOOKCHIN: The Linguistic String Parser. In: *Proc. of the* 1973 *National Computer Conference*, pp. 427–434. AFIPS, Montvale, New Jersey (1973).
[4] N. SAGER: Syntactic formatting of science information. In: *Proc. of the* 1972 *Fall Joint Computer Conference*, pp. 791-800. AFIPS, Montvale, New Jersey (1972).
[5] Z. S. HARRIS: The elementary transformations. In: *Transformations and Discourse Analysis Papers* 54, 1964. University of Pennsylvania, Philadelphia, Pa.; also in HARRIS: *Papers in Structural and Transformational Linguistics*, pp. 482–532. Reidel, Dordrecht-Holland (1970).
[6] B. B. ANDERSON: Transformationally based English strings and their word subclasses. *String Program Reports* 7. New York University Linguistic String Project, New York (1970).
[7] R. P. STOCKWELL, P. SCHACHTER and B. H. PARTEE: *The Major Syntactic Structures of English.* Holt Rinehart & Winston, New York (1973).
[8] Z. S. HARRIS: The Two Systems of Grammar: Report and Paraphrase. In: *Transformations and Discourse Analysis Papers* No. 79. University of Pennsylvania, Philadelphia, Pa.; also in *Papers* [Ref. 5], pp. 612–692.
[9] D. G. HAYS: Dependency theory: a formalism and some observations. *Language* 1964, **40**, 511–525.
[10] N. SAGER and E. BASEN: The information structure of a medical subfield (in preparation).
[11] N. SAGER: The sublanguage technique in science information processing. *J. Am. Soc. Inform. Sci.* (in press).
[12] I. D. J. BROSS and D. STERMOLE: Computer-assisted discourse analysis of a jargon. *Comput. Studies in the Humanities and Verbal Behavior* 1973, **IV**, 65–76.

PHARMACOLOGY REFERENCES

(GL641) I. M. GLYNN: The action of cardiac glycosides on ion movements. *Pharmacol. Rev.* 1964, **16**, 381–407.
(BR651) E. BRAUNWALD and F. J. KLOCKE: Digitalis. *Ann. Rev. Med.* 1965, **16**, 371–86.
(LY661) A. F. LYON and A. A. DEGRAFF: Reappraisal of digitalis—Part 1: Digitalis action at the cellular level. *Am. Heart J.* 1966, **72(3)**, 414–418.
(LE711) W. LEE and W. KLAUS: The subcellular basis for the mechanism of inotropic action of cardiac glycosides, *Pharmacol. Rev.* 1971, **23(3)**, 194–261.
(LA721) G. A. LANGER: Effects of digitalis on myocardial ionic exchange. *Circulation* 1972, **46**, 180–187.
(RO721) J. ROBERTS and G. KELLIHER: The mechanism of action of digitalis at the subcellular level. *Seminars in Drug Treatment* 1972, **2(2)**, 203–219.

APPENDIX 1
LIST OF SPECIALLY TREATED WORDS IN CORPUS

(a) *Transparent binary connectives*

along with
and
and therefore
*apo (for *appositive*)
as well as
as with
both
but
but also
*called
colon
et (as in Glynn *et al.*)
*for example
*ie
in addition
*namely
neither nor
or
other than
*paren (for *parenthetical expression*)
*particularly
*referred to as
rather than
*such as
*that is
to (as in *from 5 to 10 mm*)
+

*These words are like *be* in that the important relation is between the two arguments, rather than between argument and operator.

(b) *Transparent one-place operators*

Modals	*Aspectuals*	*Negatives*
able	achieve	never
can	appear	no
could	become	non
ile (from *contractile*)	begin	not
may	capable	un (the prefix)
might	dispose to	
must	helpful	
need	in order to	
should	in position to	
will	manifest	
would	occur	
	onset	
	process	
	property	
	seem	
	state	
	take place	
	tend	
	tendency	
	there is (like *exist*)	
	useful	

(c) *Operators which are IGNORED*

Be-like operators NOT clustered	*structural operators* NOT clustered	
be	compose	locate
characterize	consist	of
identity	containv (contain as a verb, distinct from *content*)	portion
include	found ⟨at⟩, ⟨on⟩, ⟨inside⟩, ⟨in⟩	lack
	from	within
	have	
	in	
	lack	

Relative pronouns
NOT clustered
as
that
wh
what
when
where
which
who
whose

Subordinate conjunctions
NOT clustered

after
although
as long as
because
before
er er
er than
except
for example
if then
in terms of
prior to
separately from
since
so that
under
until
upon
while
whereas
without

(d) *Middle verbs*

(when these verbs occur with one argument, argument is taken as second argument)

augment
change
decrease
diminish
improve
increase
maintain
reduce
slow

APPENDIX 2

CLUSTERS from data of 11.13.74, $t = 0{\cdot}250$
(Words truncated to 20 characters)

7.1 oubain
strophanthidin 3 bro
strophanthidin
CG
drug
compound
digitalis

7.2 drug
strophanthidin 3 bro
strophanthidin
CG
compound
erythrophleum alkalo
digitalis

7.3 strophanthidin 3 bro
oubain
CG
drug
compound
erythrophleum alkalo
digitalis

7.4 strophanthidin
ouabain
CG
drug
compound
erythrophleum alkalo
digitalis

5.1 strophanthidin 3 bro
strophanthidin
cardiotonic glycosid
CG
digitalis

5.2 strophanthidin
inhibitor
cardiotonic glycosid
CG
digitalis

5.3 strophanthidin 3 bro
cardiotonic glycosid
ouabain
cg
digitalis

5.4 strophanthidin
cardiotonic glycosid
ouabain
CG
digitalis

5.5 strophanthidin 3 bro
cardiotonic glycosid
CG
compound
digitalis

5.6 strophanthidin
cardiotonic glycosid
CG
compound
digitalis

5.7 strophanthin
ouabain
CG
drug
digitalis

5.8 Na^+
glucose
ion
sodium
calcium

5.9 Na^+
ion
sodium
calcium
potassium

5.10 turnover

intra
move
concentrate
flow

5.11 influence
stimulate
concentrate
affect
act

5.12 influence
stimulate
concentrate
affect
inhibit

5.13 similar
demonstrate
cause
due to
relate

5.14 influence
concentrate
act
affect
inhibit

5.15 demonstrate
similar
cause
relate
produce

5.16 induce
act
cause
produce
affect

5.18 stimulate
concentrate
act
affect
inhibit

4.1 sodium
Ca^{++}
Ca
calcium

4.2 cardiotonic glycosid
CG
drug
digitalis

4.3 reverse
influence
concentrate
affect

4.4 influence
induce
cause
affect

4.5 K
sodium
calcium
potassium

4.6 influence
cause
produce
affect

4.7 interfere
induce
produce
affect

4.8 induce
interfere
affect
act

4.9 induce
concentrate
affect
act

3.1 oppose
diverge
similar

3.2 agent
inhibitor
CG

3.3 atrium
heart muscle
muscle

3.4 stimulate
influence
reduce

3.5 Na + K + ATPase
enzyme
ATPase

3.6 extra
intra
move

3.7 reduce
influence
affect

3.8 Ca
ion
calcium

3.9 Ca
calcium
potassium

3.10 alter
induce
affect

3.11 ventricle
heart muscle
muscle

3.12 penetrate
concentrate
affect

3.13 increase
augment
improve

3.14 similar
link
relate

3.15 fiber
cardiac
protein

3.16 effect
produce
affect

3.17 calcium
potassium
electrolyte

3.18 decrease
increase
change

3.19 link
due to
relate

3.20 link
relate
produce

3.21 interfere
affect
toxic

3.22 due to
relate
produce

3.23 actomyosin
cardiac
protein

3.24 affect
act
toxic

2.1 ADP
El

2.2 trigger
augment

2.3 potential
species

2.4 SR
sarcoplasmic reticul

2.5 report
observe

2.6 dissociate
relate

2.7 measure
decrease

2.8 excite
depolarize

2.9 contain
lose

2.10 exchange
transport

2.11 K+
ion

2.12 space
milieu

2.13 take
treat

2.14 fold
slide

2.15 myocardium
cell

2.16 substance
ion

2.17 distribute
intra

2.18 depress
mechanism

2.19 correlate
relate