NETWORKS OF BINARY RELATIONS IN NATURAL LANGUAGES. ELECTRICAL ANALOGUES

Adrian Birbănescu

1. INTRODUCTION.

It is an essential characteristic of natural languages that one word can be concatenated with certain others to form a string that enters in correct phrases of the language, while it cannot be concatenated with others. The same holds true for strings of words. Such concatenable elements are also "mutually compatible elements" in the sense used by KVAL in a paper describing an algorithm for forming maximum classes of such elements. Mathematically, a set of ordered pairs of such "mutually compatible elements" forms a <u>relation</u>. Every string of words belonging to a language can be regarded as being obtained by successive concatenation of ordered pairs of mutually compatible elements, i.e. as formed by successive concatenation of elements of binary relations belonging to the language. Some of these string are the phrases of the language. It is thus possible to define a grammar of relations and generate by it all phrases of the language. If we try to describe by a graph this generation, the graph will be a network describing the whole system of language under consideration.

The equivalence between a grammar of relations and an IC-grammar and the equivalence between a grammar of relations and a categorial grammar are almost self-evident.

By using the notations for union, intersection and Cartesian product it is possible to write one single formula, however cumbersome, containing all phrases belonging to the language under consideration. This formula can also be interpreted as describing an electrical network which will be the electrical analogue of the language.

Let L_1 and L_2 be two languages. Let $N(L_1)$, resp $N(L_2)$ be the electrical networks assigned to L_1, resp L_2. It is possible to devise a system of electrical connections between $N(L_1)$ and $N(L_2)$ such as to obtain electrically the translation of a phrase belonging to L_2. Let's call this method "analogue translation". Because of the great number of elements involved, the construction of a complete system for analogue translation may be impractical. However, the construction of partial networks for simulating translation of a limited number of phrases may prove itself useful for demonstrational purposes.

2. DEFINITIONS 1. Let V be a <u>vocabulary</u>, that is a set the elements of which are words. Associated with the vocabulary is a operation called <u>concatenation</u> which consists of writing one or more words $a_1, a_2 \ldots a_k$ one after another. The resulting sequence $a_1 a_2 \ldots a_k$ is a <u>string</u>. By extension we shall call string also a sequence containing one single word. The empty word λ is characterized by $a_i \lambda = \lambda a_i = a_i$ for every $a_i \in V$. Some strings will be called <u>sentences</u>.

The set of all sentences generated on V is by definition the <u>language L</u>. By <u>grammar G</u> we shall understand a set of rules by which it is possible to generate the language L. Let the set of rules consist of the following :

- lists classifying all words and strings into sets called <u>categories</u> ;
- rules of the form

$$\mathcal{R} = \bigcup_{i=1}^{n} \mathcal{A}_i \times \mathcal{B}_i \qquad (1)$$

where $\mathcal{R}$, $\mathcal{A}_i$ and $\mathcal{B}_i$ (i=1...n), are categories, and the ordered pair (a_{ij},b_{ik}), for $a_{ij} \in \mathcal{A}_i$; $b_{ij} \in \mathcal{B}_i$ is associated to the string a_{ij} b_{ik}. We admit that any $\mathcal{A}_i$ or $\mathcal{B}_i$ may contain one single word, or even only the empty word λ;

- a list of the categories which are sentences.

3. EXAMPLE 1. Let the grammar G_R be defined by :

- the vocabulary

$V = \{$ poor, dear, John, Richard, sleeps, reads, $\lambda \}$;

- the categories

$\mathcal{C} = \{$poor, dear$\}$

$\mathcal{D} = \{$John, Richards$\}$

$\mathcal{B} = \{$sleeps, reeds$\}$

- the rules

$\mathcal{A} = (\mathcal{C} \cup \lambda) \times \mathcal{D}$

$\mathcal{S} = \mathcal{A} \times \mathcal{B}$

- the list of sentences containing only $\mathcal{S}$.

In this simple case, the rules are of form (1), with i=1. Now, starting from $\mathcal{S}$, by successive substitutions we obtain

$$L(G_R) = \mathcal{S} = \mathcal{A} \times \mathcal{B} = [(\mathcal{C} \cup \lambda) \times \mathcal{D}] \times \mathcal{B} =$$
$$= [\{\text{poor, dear}, \lambda\} \times \{\text{John, Richard}\}] \times \{\text{sleeps,reads}\} \quad (2)$$

The set $\mathcal{S}$ is thus composed of 12 ordered pairs and triplets. Writing down the strings associated to these pairs and triplets we enumerate the sentences of the language L(G_R) generated by G_R :

poor John sleeps
poor John reads
poor Richard sleeps
poor Richard reads (3)
dear John sleeps

dear John reads

dear Richard sleeps

dear Richard reads

John sleeps

John reads

Richard sleeps

Richard reads

4. REMARKS 1. A set like $\mathcal{A} = (\mathcal{E} \cup \lambda) \times \mathcal{D}$ is by definition a binary relation on V. Similarly, if V^2 is the set of all strings obtained by the concatenation of two words ("strings of length 2"), then $\mathcal{S}$ is a binary relation on $V^2 \cup V$. These relations have direct linguistic interpretations, for $\mathcal{A}$ may be reguarded as the relation between adjective and noun , while $\mathcal{S}$ is the relation between noun group and verb. Of course, these simple interpretations are valid within the limited grammar exposed above.

For convenience of description, in what follows we shall call a grammar of the type defined in 2, e.g. G_R in example 1, a grammar of relations.

Sets like $\mathcal{A}$ and $\mathcal{S}$ in example 1 are sets of ordered pairs of strings whose concatenation leads to other strings that can belong to sentences of the language under consideration. Concatenable elements are also compatible elements, by compatibility understanding a symmetric nontransitive relation. Regarded as such, these elements can be classified into classes, one of which is maximal, by means of an algorithm developed by KARLGREN [1] .

We are interested to classify concatenable elements by imposing the restriction, that follows.

5. DEFINITION 2. Two categories $\mathcal{A}_1$, $\mathcal{A}_2$ are called different if there is at least one third category $\mathcal{B}$ such that:

a) either

a_1b is a string contained in at least one sentence of the language, i.e. a_1b belongs to a category of the language, for every $a_1 \in \mathcal{A}_1$, $b \in \mathcal{B}$, while a_2b is contained in no sentence of the language, which ever would be

$a_2 \in \mathcal{A}_2$, $b \in \mathcal{B}$;

b) or

a_2b is a string contained in at least one sentence of the language, for every $a_2 \in \mathcal{A}_2$, $b \in \mathcal{B}$, while a_1b is contained in no sentence of the language, which ever would be $a_1 \in \mathcal{A}_1$, $b \in \mathcal{B}$.

6. EXAMPLE 2. Let's consider the following relations

$\mathcal{R}_1 = \mathcal{A}_1 \times \mathcal{B}_1$

where $\mathcal{A}_1 = \{$ this, that $\}$

$\mathcal{B}_1 = \{$ girl, boy, man, woman, aunt, ... $\}$

$\mathcal{R}_2 = \mathcal{A}_2 \times \mathcal{B}_2$

where $\mathcal{A}_2 = \{$ these, those $\}$

$\mathcal{B}_2 = \{$ girls, boys, men, women, aunts, ... $\}$

According to definition 2, $\mathcal{R}_1$ and $\mathcal{R}_2$ are of different categories since there exist in English categories

$\mathcal{C}_1 = \{$ sees, wants, comes, reads, sleeps, ... $\}$

$\mathcal{C}_2 = \{$ see, want, come, read, sleep, ... $\}$

such that

$\mathcal{R}_1 \times \mathcal{C}_1$, $\mathcal{R}_2 \times \mathcal{C}_2$ are categories, (also relations), of the English language, while

$\mathcal{R}_1 \times \mathcal{C}_2$, $\mathcal{R}_2 \times \mathcal{C}_1$

are not. It is interesting to note that $\mathcal{R}_1 \times \mathcal{C}_1 \cup \mathcal{R}_2 \times \mathcal{C}_2$ may be a relation belonging to the English grammar.

7. THEOREM. For every grammar of relations we can find an equivalent IC-grammar (immediate-constituents grammar) and conversely.

The proof follows immediately from the observation that any rule of the form (1), i.e.

$$\mathcal{R} = \bigcup_{i=1}^{n} \mathcal{A}_i \times \mathcal{B}_i$$

can be substituted by the following set of IC-grammar rules

$$R \longrightarrow D_1$$
$$R \longrightarrow D_2$$
$$\cdots\cdots$$
$$R \longrightarrow D_i$$
$$\cdots\cdots$$
$$R \longrightarrow D_n$$
$$D_1 \longrightarrow A_1 B_1$$
$$D_2 \longrightarrow A_2 B_2$$
$$\cdots\cdots$$
$$D_i \longrightarrow A_i B_i$$
$$\cdots\cdots$$
$$D_n \longrightarrow A_n B_n$$

and conversely.

At the same time, to every category $\mathcal{C}_j = \{ C_1, C_2, \ldots, C_m \}$ in a grammar of relations, corresponds a terminal rule

$$C \longrightarrow c_1, c_2, \ldots c_m$$

of the equivalent IC-grammar and conversely.

8. REMARK 2. From the equivalence between a grammar of relations and an IC-grammar it followes also the equivalence between a grammar of relations and a categorial grammar. The necessary proofs can be found in BAR-HILLEL, GAIFMAN and SHAMIR [2], and related comments in KARLGREN [3].

9. EXAMPLES 3. a) The language described in example 1 can be generated also by the following IC-grammar :

- terminal vocabulary

  $V_T = \{$poor, dear, John, Richard, sleeps, reads, $\lambda\}$

- auxilary vocabulary

  $V_N = \{S, A, B, C, D\}$

- rules

  $S \longrightarrow AB$

  $A \longrightarrow CD$

  $A \longrightarrow D$

  $C \longrightarrow$ poor, dear,

  $D \longrightarrow$ John, Richard

  $B \longrightarrow$ sleeps, reads

b) In the equivalent categorial grammar,

- poor, dear are of category n/n

- John, Richard are of category n

- sleeps, reads are of category $n\backslash s$

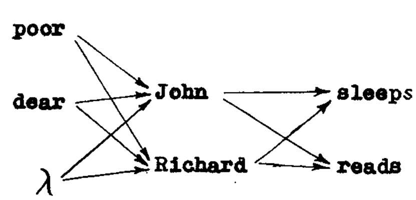
- S is the sentence category.

1o. GRAPHICAL REPRESENTATION. We shall associate to each grammar of relations a graph, observing the following conventions:

- each path must be followed from the extreme left to the extreme right, along the arrows ;

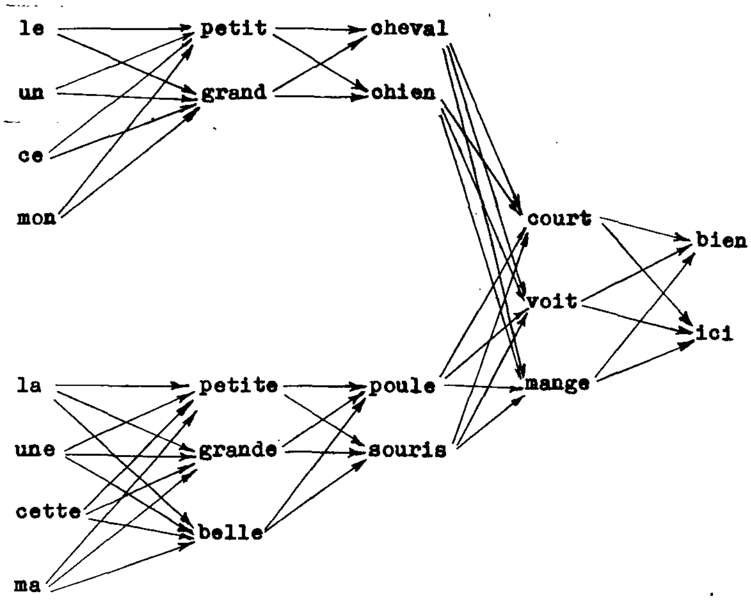
- a sentence is a sequence of words found along a path.

Thus, the graph corresponding to grammar G_R in example 1 is



(4)

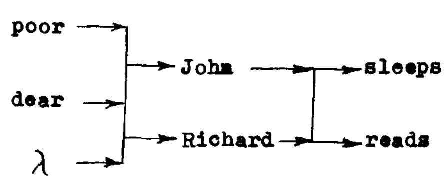
An example given by the author for the French language in [4] is



(5)

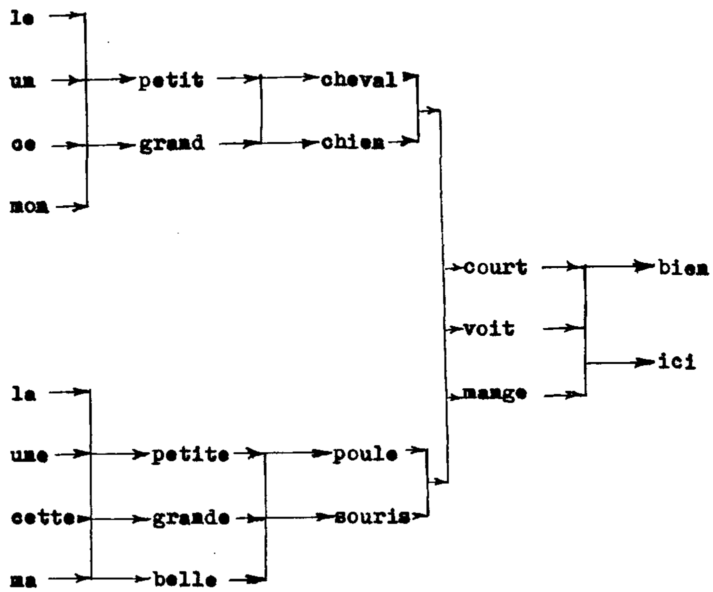
Such graphs are called networks. An earlier example is to be found in MARCUS [5] , Taking into consideration what has already been written at 4 (ie. in remark 1) , the above graphs can be regarded as <u>networks of binary relations</u>. They desoribe not only sentence structures, but also the whole system of the language,

Now we shall simplify the graph (4) without altering it topologically



(6)

In the same manmer, the graph (5) becomes



(7)

11. ELECTRICAL ANALOGUES. If a graph like (6) or (7) is considered to be an electrical diagram where each word is substituted by a contact, and each arrow by an electrical conductor (wire), the result is an electrical analogue of the grammar. In this analogue if one closes all contacts corresponding to a sentence, a continuous electrical path is established and the current flows from one extremity to the other. The detection of this current is a proof of "grammaticality" for the word sequence under consideration.

Electrical analogues can be designed also algebraically. For this it is necessary to proceed as follows :

- starting from the list of sentences $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_p$ write

$$L = \bigcup_{j=1}^{p} \mathcal{S}_j$$

- replace each $\mathcal{S}_i$ by its corresponding rule of form (1), that is

$$L = \bigcup_{j=1}^{p} \left(\bigcup_{i=1}^{n} \mathcal{A}_i \times \mathcal{B}_i \right)$$

- replace each $\mathcal{A}_i$ and each $\mathcal{B}_i$ by the corresponding rules of type (1),and so on until the right member of the formula contains only the words of the vocabulary. Such a formula is (2) in example 1' We have now at our disposal a formula enumerating the ordered n-tuples associated to all sentences of the language L. This formula is to be interpreted in terms of switching algebra as follows :

- an n-tuple $(a_1,a_2,...a_r)$ corresponds to a series connection of the elements $a_1,a_2,...a_r$;

- the union $\bigcup_{i=1}^{s} \mathcal{D}_i$ corresponds to a paralel connection of the elements $\mathcal{D}_i$;

- the Cartesian product $\mathcal{A}_i \times \mathcal{B}_i$ corresponds to the parallel connections of all series connections $a_i b_i$, where $a_i \in \mathcal{A}_i$, $b_i \in \mathcal{B}_i$

In [4] is presented an electrical analogue of the grammar described by graph (7).

12. GRAPHS OF TRANSLATIONS. This chapter and the following are intended as suggested applications of the above discussion, to the understanding of the process of translation. No attempt is made to start from more rigorous definitions, as may be found for example in [6] or [7] . Here the process of translation of a simple sentence from language L_1 , into language L_2, is regarded as consisting of the following operations :

a) seek the given sentence in the dictionary L_1-L_2 ;

b) if the whole sentence is found in the dictionary, write down the translation found there and end the process ;

c) if the sentence is not found in the dictionary, divide it into two substrings admitted by the grammar (in fact, <u>immediate constituents)</u> ;

d) seek each of the substrings in the dictionary ;

e) if one substring is found in the dictionary, write down its translation as given in the dictionary ;

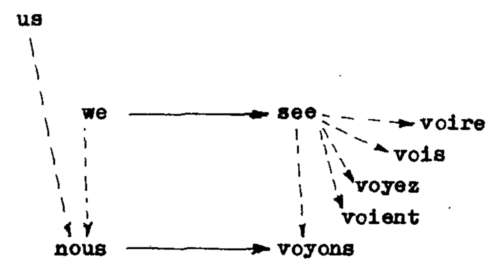
f) if one substring is not found in the dictionary, divide it again into two further substrings and then proceed again as indicated under d) ;

g) the process stops when a string is obtained which contains only words belonging to the vocabulary of the language L_2.

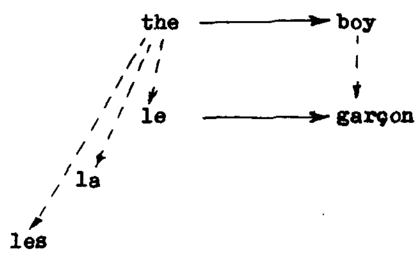
A difficulty rises currently during the process of translation, and this is due to the fact that many words, or strings composed of more than one word , admit two or more translations.Such is the case with homonyms. Special subroutines have been developed to solve the problem of homonyms in digital translation of language. Such subroutines are based on successive steps of conditioned decision To quote only a few very simple examples, MARCUS , in [8],gives sketches of algorithms for translating into Roumanian "example" and "this "and for solving the homonymy of the French "pas" or the English "this". The problem is related to that of sequential understanding of a sentence, as described by ZIERER [9] .

Let us put the problem somewhat differently. To choose the write word (or substring) between more than one possible variants, we need some supplementary condition, or conditions. A first and most important condition is that the right word (substring) must match grammatically the other words (substrings) in the string. That is to say that the right word (string) must form with another word (substring) in the string an ordered pair belonging to a certain relation accepted by the grammar of the language L_2. In the graphical representation suggested above, the right word (substring) must find itself on a continuous path with the other words (substrings) contained in the translation of the sentence under consideration.

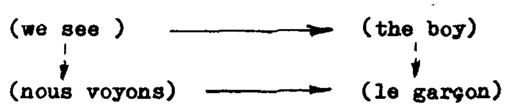
For example, let L_1 be the English language and L_2 the French. Let the sentence in L_1 be "we see the boy". It must be divided into (we see) (the boy). Putting side by side the corresponding parts of the English and French graphs, and marking by dotted lines the mapping defined in the dictionary, we can draw for the first substring the graph



We choose as translation only that substring that closes our diagram. Similarly, for the second substring



The final translation is given by



There are cases when the condition of grammaticality is not sufficient. Then a human translator uses supplementary information, like general knowledge of the subject treated in the text, style used etc. Graphically, such information may be taken into account, for example, by assigning different colours to different types of subject. Then the diagram must close through paths of different colour.

13. ELECTRICAL TRANSLATION. Let us assume now that we have at our disposal an electrical analogue $N(L_1)$ of the language L_1, and an electrical analogue $N(L_2)$ of the language L_2. We can further immagine such an electrical connexion between the two analogues that when a contact "a_i" closes in $N(L_1)$, all contacts corresponding to the different possible translations of "a_i" indicated by the dictionary L_1-L_2 are closed in $N(L_2)$. Then, when a continuous path of contacts is closed in $N(L_1)$, its translation can be only a continuous path resulted in $N(L_2)$. For the selection of the type of subject or of the style used, we can devise a switch that makes only the corresponding connections between $N(L_1)$ and $N(L_2)$, or in $N(L_1)$ and $N(L_2)$ themselves. Such a switch may have, for example, positions marked : literature, mechanics, electricity, electronics, chemistry, medicine etc.

Some texts may contain sufficient information to enable the above switch to find automatically its right position. This idea deserves an entirely separate discussion.

14. CONCLUDING REMARKS. What was suggested under 13 and 14 are in fact examples of <u>cabled logic</u>. The implementation of these ideas for an entire language may encounter tremenduous technical difficulties. Designing graphs and electrical analogues for limited parts of language may prove however useful for demonstrational purposes. Thus it would be possible to achieve other models of language understanding and translation than those provided by digital programs and computers. The author feels that this is indeed interesting, for as explained by ZIERER [9] , the process of understanding must not be actually as divided into elementary steps as in an algorithm : "Dazu kommt noch dass beim Verstehen der gesprochenen Sprache durch den Menschen auch der soziale Kontext und das Erlebnisvermögen des Menschen zum Abbau der angebotenen Information beiträgt.

Hierin ist der Computer dem Menschen unterlegen".

Another author, SAUVAN [1o] , discussing other subjects, believes that a sequential computer cannot treat adequately a combinatorial problem for it has no possibilities of global perception: "L'auteur est persuadé que les recherches doivent s'orienter vers les logiques câblées semblables aux structures cérébrales".

BIBLIOGRAPHY

1. HANS KARLGREN, An algorithm for forming a maximum class of mutually compatible elements out of a partially incompatible set, KVAL PM 385, 25.1.68, Research group for quantitative linguistics, Stockholm.

2. Y.BAR-HILLEL, H.GAIFMAN and E.SHAMIR, On categorial and phrase structure grammars, Bull.Res.Council of Israel,9 F,196o, p.16.

Also to be found in

YEHOSHUA BAR-HILLEL, Language and Information , Addison-Wesley Publishing Company & The Jerusalem Academic Press, Reading / Massachussetts - Jerusalem 1964.

3. HANS KARLGREN, Categorial grammar analysis of context-sensitive languages, KVAL Ref. No.441, 1968, Research group for quantitative linguistics, Stockholm.

4. ADRIAN BIRBANESCU, Modèle graphique et modèle électrique d'une grammaire à constituants immédiats. Cahiers de linguistique théorique et appliquée, V, 1968, Bucharest.

5. SOLOMON MARCUS, Introduction mathēmatique à la linguistique structurale, Paris, Dunod, 1967.

6. KAREL ČULIK, Semantics and translation of grammars and ALGOL-like languages, Kybernetika (Czechoslovakia), 1, nr.1, 1965, p.47-49.

7. JIŘÍ KOPŘIVA, Generalization of well-translation of formal languages. Kybernetika (Czechoslovakia). 2, nr.4,1966,p.3o5-313.

8. S.MARCUS, Noţiunea de algoritm şi implicaţiile ei lingvistice,Limba română, X, nr.3,1961,p.2o4-217.

9. ERNESTO ZIERER, Kybernetik und Sprachverstehen. Grundlagenstudien aus Kybernetik und Geisteswissenschaft, lo,nr.1,mart 1969, p 11-14.

lo. JACQUES SAUVAN,Informatique Combinatoire et méta-informatique,Cybernetica (Namur), nr.4, 1968,p.229-234.