OPTIMIZATION ALGORITHMS OF DECIPHERING AS THE ELEMENTS OF A LINGUISTIC THEORY

B.V.SUKHOTIN
Institute of the Russian Language
121 019, Volkhonka 18/2, Moscow, USSR

Abstract

This paper presents an outline of the linguistic theory which may be identified with the partially ordered set of optimization algorithms of deciphering. An algorithm of deciphering is the operational definition of a given linguistic phenomenon which has the following three components: a set of admissible solutions, an objective function and a procedure which finds out the minimum or the maximum of the objective function.

The paper contains the description of the four algorithms of the proposed type:
1. The algorithm which classifies the letters into vowels and consonants.
2. The algorithm which identifies the morphemes in the text without the boundaries between words.
3. The algorithm which finds out the dependency tree of a sentence.
4. The algorithm which finds out the mapping of the letters of an unknown language into the letters of a known one.

The forties and the first half of the fifties were marked by the pronounced interest of the linguists to the so-called "discovery procedures". These investigations were not very successful at that time. The Chomskyans' criticism also hindered the progress in this direction.

There is no reason to revive the old discussions. We will try to show further that the optimization algorithms we propose combine the theoretical generality on the one hand with the practical usefulness on the other. Moreover it appears that the methods of the generative grammar theory and those of the discovery procedures are even not at all contradictory. For example, in a recent work of M.Remmel the set of the admissible solutions is determined as a set of the generative grammars of N.Chomsky.

In this paper we prefer to use the term "deciphering procedures (algorithms)" instead of "discovery procedures", because the latter implies the operations which are not necessarily formal.

An algorithm of linguistic deciphering is a formal procedure aimed at the recognition of linguistic objects in a text whose language is not known to the investigator.

Assuming that any deciphering procedure may serve as a definition of the respective linguistic object we may vew the set of such procedures as a certain linguistic theory which has the following properties:

1) A great degree of generalization, because its definitions should be valid both for the known and unknown languages.

2) Formality, because naturally enough, the deciphering procedures should be presented in the shape of algorithms.

3) Constructivity, i.e. the possibility of identifying a certain linguistic object with the help of a deciphering procedure within a reasonable time interval.

To identify a linguistic object a deciphering algorithm makes use of a set of its features. It seems obvious that a linguistic object cannot be defined by means of binary features alone. The following scheme seems to be better founded:

I. Binary features are used to determine the general type of certain linguistic objects. The objects belonging to that type form the set of admissible solutions of a deciphering problem.

2. An objective function which estimates the quality of each solution is introduced on the set of admissible solutions. The values of the objective function are calculated with the help of the investigated text. They reflect the individuality of the given language. A maximum or a minimum of the objective function should correspond to the linguistic object which is to be defined.

3. It follows that a deciphering procedure should be an optimization algorithm which finds "the best" admissible solution - from the point of vew of the objective function.

Thus, the set of admissible solutions, the objective function and the optimization algorithm constitute the definition of a linguistic object which may be used for the purposes of deciphering; a definition of this kind will be further referred to as a deciphering algorithm, or simply, an algorithm.

There is a natural hierarchy of deciphering algorithms. An algorithm B is senior to an algorithm A if the former makes use of the information provided by the latter. If A and B work alternatively each time improving the output, then the seniority is determined by the first iteration. Taking into account the fact that the set of essentially different algorithms should be finite, it appears that there must exist "zero" algorithms which use no information produced by any other deciphering algorithms.

Zero algorithmz should be different due to the fact that the physical substances of different languages may be different too. Thus the zero algorithm for the analysis of the written form of languages should be able to discriminate between a dark spot and a light one and to identify the place of each spot on the page; it should discover the set of alphabetic symbols of the language. A similar algorithm adjusted to the analysis of audible speech should produce the alphabet of phonemes, exploiting its capacity to discern certain minimal differences of sonation. The plurality of zero algorithms may be reduced by converting signals of different nature into a set of curves. As it is well known such algorithms are the goal of pattern recognition theory.

Senior algorithms should be used for the analysis of grammar; the highest levels correspond to the problems of semantics and translation.

Many algorithms of different levels display great similarity and sometimes even identity, their only difference consisting in the linguistic material which serves as the input. The following types of the algorithms may be pointed out:

I. Algorithms of classification, which divide the set of investigated objects

into several subsets.

2. Algorithms of aggregation which form larger units from smaller ones.

3. Algorithms of connection which find out some relation of partial ordering.

4. Algorithms of mapping the elements of an unknown language into the elements of a known one.

The most simple classification algorithm is that which classifies the set of letters $A = \{l_i\}$ into vowels and consonants. In this case an admissible solution is a division

$$D = \{V, C\}, \quad V \cup C = A, \quad V \cap C = \emptyset$$

The objective function reflects the fact that letters of the same class co-occur rather rarely whereas letters of different classes co-occur relatively more often; it is formulated as follows:

$$Q(D) = \sum_i \sum_j f(l_i, l_j), \quad l_i \in V, \quad l_j \in C$$

Here $f(l_i, l_j)$ denotes the frequency of letters l_i and l_j. The maximum of Q(D) corresponds to the optimal classification. An appropriate optimization procedure reduces the amount of divisions that should be evaluated to a reasonable number. This algorithm has been thoroughly tested in a number of computer experiments and in every case yielded almost entirely correct results.

The most important algorithm of aggregation is the morpheme identification algorithm. Apart from identifying morphemes this algorithm discovers an IC graph which shows the way in which morphemes are combined into words. An admissible solution in this case is a sequence of divisions $D_I, \ldots, D_n$ of the text, each class of D_{i+1} being included in a certain class of D_i. A morpheme m is the string of letters at least one occurrence of which should be an element of a certain class of D_i.

The sequence $D_I, \ldots, D_n$ determines the set of morphemes in a unic way. The objective function is set up by ascribing to each morpheme a certain number q(m) which is great when m consists of the letters which predict each other stronger than they predict the letters of the neighbouring morphemes. A number of experiments have been carried out; the best results have been obtained with the help of the following function:

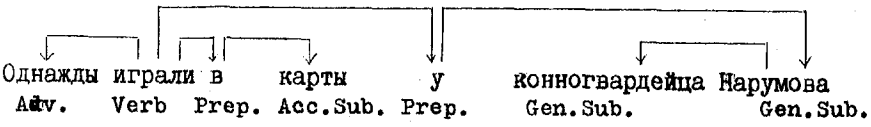
$$q(m) = q(aXb) = \frac{f^2(aXb)}{\max(f(aX), f(Xb))} - \max_{x,y}(f(aXbx), f(yaXb))$$

Here f denotes the frequency of a string, a is the initial, b is the final letter of m, y is a letter which precedes m, x is a letter which follows it, X is a string. The best solution should correspond to the maximum of $Q(M) = \sum_i q(m_i)$, where $M = \{m_i\}$. A Russian text of 10000 letters was chosen for the experiments. Here is an extract of the analysed text:

((челове)к) с ((лопатк)ой) (горьк) ои (укоризн)(енно) усмех ((нул)ся)

Representative of the algorithms of the third type is the algorithm of finding the dependency graph of a sentence. For this purpose the words of the language should be classified into syntactical classes so that we may consider a word v to be included in a class K_v. The conditional probability $P(K_v/K_w)$ of occurrence of K_v near K_w is calculated with the help of the text.

The set of admissible solutions is the set of all possible dependency trees which may be ascribed to a given sentence. The conditional probabilities provide the weights for the arcs of the tree. The quality of a tree is the sum (or the mean) of the weights of all arcs. The optimal tree presumably has the maximum quality. A great number of the algorithms of this type have been tested in computer experiments; the best ones correctly identified more than 80% of connections. Here is a typical example taken from an experiment which was carried out for a Russian text of 10000 words:



Algorithms of this type may be used for the purposes of machine translation, in which case a greater amount of the input information is needed.

A typical example of an algorithm which obtains the mapping $M = \{E_i \rightarrow E_i'\}$ (E_i being some elements of the unknown language, E_i' - the respective elements of the known one) is furnished by the algorithm which discovers the pronunciation of letters.

It is based on the hypothesis that letters of two different languages which have similar pronunciation possess similar combinatory power as well.

The combinatory power of the letter l_i may be described by the vector of conditional probabilities $C_i = P(l_i/l_x)$ which characterizes the occurrences of l_i in the neighbourhood of l_x. In the same way, the vector $C_i' = P(l_i'/l_x')$ characterizes the combinatory power of l_i'.

The quality of a mapping may be estimated by the formula:

$$Q(M) = \sum_i d(C_i, C_i') = \sum_i d(l_i, l_i')$$

Here d denotes the distance (e.g. Euclidean) between the vectors C_i and C_i'. All pairs $l_i \rightarrow l_i'$, $l_x \rightarrow l_x'$ belong to the mapping M, so that d may be calculated by the formula:

$$d(l_i, l_i') = \sqrt{\sum_x (P(l_i/l_x) - P(l_i'/l_x'))^2}$$

The minimum of Q(M) corresponds to the optimal mapping. Some algorithms of this type have been tested with interesting results. It is obvious that a similar algorithm will be able to compile a bilingual dictionary with the entries in the unknown language, although the latter problem is, naturally, far more difficult.

References

Сухотин Б.В. (1962). "Экспериментальное выделение классов букв с помощью ЭВМ". Проблемы структурной лингвистики, М.

Сухотин Б.В. (1975) "Оптимизационные алгоритмы лингвистической дешифровки". НТИ, сер.2, №5.

Сухотин Б.В. (1976) "Оптимизационные методы исследования языка". М.

Сухотин Б.В. (1984) "Выделение морфем в текстах без пробелов между словами". М.

Гиндилис Л.М., Каплан С.А., Кардашев Н.С., Пановкин Б.Н., Сухотин Б.В., Хованов Г.М. (1969) "Внеземные цивилизации". М.