[Presented at the Conference on Mechanical Translation
at the Massachusetts Institute of Technology, 18 June 1952. Not published.]

Operational Grammar

Y. Bar-Hillel

The purpose of this talk is to elaborate on one of the points treated in my report "The present state of research on mechanical translation"[1], preprinted copies of which have been sent to you some time ago. I said there a few things about the necessity of developing what I called an Operational Grammar (or Syntax). I think that this topic is important enough to warrant an expanded treatment.

Since the conception of an Operational Grammar is rather unfamiliar, let me start with a situation well known to all of you and sufficiently analogous to the situation we are facing in MT to be of help in my exposition. A student of chemistry who had never before taken a course in Analytical Chemistry would be at a complete loss if presented with the task of finding out the composition of a given mixture of substances in a test-tube, even if he were to know all properties of all chemical substances by heart. If endowed with exceptional intelligence, he might perhaps be able to work out a method of analysis for himself, within a few years of uninterrupted thinking. But this is, of course, not the standard procedure. The student is simply told, in a special course, what operation to perform first on the content of the test-tube, and then, depending on the outcome of this operation, what to do next, and so on. He may be told – I forget already what I myself was taught 20 years ago, but it does not matter – to put first the whole mixture into hot water. As a second step, he will be instructed to continue in a certain way if everything dissolves, and in some other way, if not everything dissolves. In the latter case, he will probably be told to filter the stuff and then ....

I am not sure whether it can be shown that the standard procedure which the chemistry student is taught in qualitative Analytic Chemistry is, on some kind of average, an optimal procedure, with respect to some measure that would take into account, among other factors, also time and cost. But it is probable that this procedure is at least a good one, because it seems likely that otherwise the one or the other of the analytical chemists would have found a better one.

The situation with regard to grammatical analysis seems to me to be fairly analogical. A student of German, even if he were to know all of descriptive German Grammar, but without having taken a special course in operational Grammar (Analytic Grammar would have been a better term, but this expression is already in use in a different sense), would be in a fix if presented with the task of finding out the grammatical structure of a given German sentence. He might well be stuck with his very first step. Where should he start his analysis? Nothing in what he knows about German grammar compels him in a univocal way to start at some point and not at another. As a matter of fact, language teachers, especially the good ones, would give their students some hints, which [would] not, however, form a complete and infallible system and, what is as bad for our purpose, would teach them to rely on semantic shortcuts using their understanding, or at least partial understanding, of what the material under investigation is about. Hence it is understandable why so far no Operational Grammars of any language have been written, to the best of my knowledge, whereas there are thousands of textbooks on Analytic Chemistry. There was indeed no such great practical urgency for an Operational Grammar as there was for an Analytic Chemistry textbook. But I would say that, from a theoretical standpoint, especially that of Structural Linguistics, the construction of an Operational Grammar is of greatest importance. It is, I would say, a challenge that must be faced by modern linguists. But, in addition to that, it is of paramount value for MT. No

[1] American Documentation, vol. II, no. 4, 1952 [actually dated 1951 but printed in 1953, Editor]

semantic shortcuts will stand at the disposal of the translation machine – at least not for the near future –, and therefore only a very elaborate full-fledged sequentially programmed instruction system will enable the machine to carry out the grammatical analysis of any given sentence.

The preparation of such a system is difficult but not too difficult. The main problem is not in finding *a* procedure but in finding a *good* procedure, though not necessarily the best one, again according to some measure involving time and cost. To start with, any system would do, even an inefficient one, even an incomplete one. Improvements will then be made almost automatically, simply through the failures of the machine itself. I think that a good linguist with a staff of a few assistants and clerks should be able to provide such a provisional system for any language that has already been more or less exhaustively described –like English, German, or Russian – within a year or two.

With regard to German, for instance, a combination of the methods proposed by Pollard[2] and Oswald-Fletcher[3] should be sufficient to give us a good start towards the preparation of an Operational Grammar for German. Following Pollard, the first operation to be carried out on a symbol-sequence recognised as a unit of analysis might be the count for the number of commas in this unit. It is obvious that this is something machines could easily do. If the unit contains no commas, then the next operation may consist in checking whether the last word of the unit is a noun (and it does not matter whether this is done by inspection of its first letter and recognition of its being or not being capitalised or by looking up this word in a unilingual word-category list in which this last word would be listed as a noun or otherwise). If the last word turns out to be a noun, then the next operation would be to check whether the unit contains a so-called participial construction. This is a rather complicated procedure, and I shall not go into the details here. If it turns out that no such construction occurs, the next instruction might well be not to bother any more about grammatical analysis, and this because under all these circumstances the German word-order will now correspond, in general, to some standard English word-order, so that the translation should not provide any special difficulties of a syntactical nature. Should any of the outcomes of any of the mentioned steps be different from that assumed in our example, different instructions will have to be followed.

In the given example, I followed more or less Pollard's lead. But Pollard assumes far too much for our purposes. Many of the things he assumes, rightfully, that an intelligent and somewhat experienced human student will be able to determine for himself quickly, almost automatically, have to be incorporated explicitly and painfully into what the machine will have to do.

Sometimes, the machine will come out with a multiple syntactical resolution of the given unit. If the given sentence were – to take now an English example – <u>John thought that Paul lied</u>, then there is nothing *in* this sentence, in its printed form, that would exclude an interpretation different from that presumably intended by its producer. <u>that</u> could be a demonstrative (<u>that Paul</u> against <u>this Paul</u>), though such an interpretation would hardly fit in the great majority of the situations where a token of this sentence is produced. Since I do not envisage that the translation machine of the near future will be able to take into account, for its grammatical analyses, larger contexts than those between periods (and equivalent border-signals), I think that the best procedure would be to have the machine present both resolutions and to leave it to the post-editor to eliminate the unsuitable one, which in this case would probably be very simple.

To arrive at the (or a) resolution of a given text, the machine might well have to perform thousands, perhaps hundreds of thousands, of elementary operations. But since these

---

[2] *A Key to Rapid Translation of German*. University of Texas, 1947

[3] V. A. Oswald, Jr. and S. L. Fletcher, Jr: Proposals for the Mechanical Resolution of German Syntax Patterns, *Modern Language Forum* 36, no. 3-4, 1951.

elementary operations – nothing more than simple matchings – can be performed at a speed of a few microseconds per operation, the huge number need not impress us too much.

Let me now point out an approach which by itself, or in combination with those I mentioned before, might advance us even more. Oswald-Fletcher work with a certain number of word-categories, I think 92. These are denoted by the numerals ‘1’, ... ‘92’. They are working with instructions of the type “in a sequence of ... words belonging, in order, to categories ..., is the direct object of the clause of which it is a part.” Sometimes the instructions are much more complicated and qualifications starting with “unless” are often used.

I would like to show that by a simple change in the notation many, in some cases perhaps all, of the syntactic instructions can be eliminated. In their place, a simple quasi-arithmetical operation, easily performable by a machine, will yield the required results.

Since I treated this approach – due mainly to the Polish logician K. Ajdukiewicz – at length somewhere else[4], let me present here only a very simplified account of it. Assume that the syntactic category of *nominals* (i.e., approximately, the category of words and word-blocks that function syntactically like proper names) is denoted by ‘n’. Then we denote the category of those words and word-blocks that form a sentence out of a nominal – such words correspond approximately what is called, in customary terminology, intransitive verbs, – by ‘s/n’[5], or to be more specific, by *s/n* (omitting from now on quotation-marks, for the sake of simplicity), where the parentheses indicate that the nominal is to the immediate left, or by *s/[n],* where the brackets indicate "to the right". Words that out of one left and one right nominal form a sentence, (transitive verbs) are *ns/(n)[n]*. Words that out of a nominal (to the right) form a nominal (adjectives) belong to the category *n/[n]*, those that out of sentence (to the right) form a nominal to *n/[s],* etc.

I assume that a word-category-list has been prepared which gives for each English word a full list of the syntactic categories to which this word (in Peirce’s terminology, *tokens* of this word) might conceivably belong. I shall now show you how in an admittedly very simple, and moreover simplified, case a machine would perform a syntactic analysis. The unit to be analysed is

John thought that Paul lied.

Assume that the word-category-list shows that John is a *n*, thought is either a *n* or a *s/(n)* or a *s/(n)[n],* or a *s/(n)[s],* that is either a *n* or a *n/[n]* or a *n/[s],* Paul a *n*, and lied a *s/(n)*. Let us represent the situation in the following graphic way:

John	thought	that	Paul	lied
n	n	n	n	s/(n)
	s/(n)	n/[n]		
	s/(n)[n]	n/[n]		
	s/(n)[n]			

The "cancellation" to be performed by the machine is the replacement of the index-sequence of the form β α/(β) by α, α/[β] β by α, and β α/(β)[φ] φ by α. Whenever subsequent cancellations lead to just one final index *s*, the corresponding resolution "fits". It is easy to check that out of the twelve initially possible resolutions only two fit, viz:

(1) n s/(n)[s] n/[n] n s/(n)

[4] A Quasi-arithmetical Notation for Syntactic Description, *Language* [vol.29 (1953), 47-58; reprinted in his collection *Language and information* (Reading, Mass.: Addison-Wesley, 1964), 61-74. Editor]

[5] The original typescript represents the symbols of Bar-Hillel’s category grammar thus: $\frac{n}{n}$, $\frac{s}{[n]}$, $\frac{ns}{(n)[n]}$, etc.
In this reproduction I have substituted Bar-Hillel’s later practice of using slashes: n/n, s/[n], ns/(n)[n],etc. [as in the reprint of the article in footnote 4; Editor]

which yields, step after step,

(2) n s/(n)[s] n s/(n)

(3) n s/(n)[s] s

(4) s

(where, notice, that is treated as a demonstrative adjectival) and

(1') n s/(n)[n] n/[s] n s/(n)

which yields

(2') n s/(n)[n] n/[s] s

(3') n s/(n)[n] n

(4') s

To arrive at the result that only these two resolutions "fit", the machine may well have to perform dozens of matchings, and this for a five-word sentence, For a thirty-word sentence, we might arrive at the hundreds of thousands mentioned above. But, as I said, we need not be afraid of the large numbers.

The same procedure will, incidentally, also reveal what are the *blocks* in the given sentence. (Thereby a problem posed earlier by Bull will find its solution.) A block is simply a word sequence whose index is *single*, though not necessarily *s* or any other *simple* index.

In our example, that Paul would be. a block (a nominal) with respect to the first resolution, but not Paul lied. With respect to the second resolution, Paul lied is a block (a (sub-)sentence) but not that John. This, incidentally, shows also that the block-character of a sequence depends on the resolution of the whole sentence of which this sequence is a part. Taking up Bull's example all the very great banks, it can now be seen that a machine could easily decide that this sequence forms a (possible) block, since at least one corresponding index-sequence, viz:

n/[n] n/[n] n/[n]/n/[n] n/[n] n

(notice the index of the adverbial very!) yields *n* as its fina1 index, as you may check for yourself. I shall not discuss here the sophistications necessary to account for the differences in function and position of the *n/[n]* - words occurring in this block.

I said before that the applicability of this method is based on the assumption that a word-category-list has been prepared. This would mean, for English for instance, a list with some million and a half entries. The preparation of such a list is certainly not a simple task, since all possible occurrences of these words in all kinds of syntactic constructions have to be envisaged. Nor would it be easy to have a storage built for a machine to contain all this information with sufficiently quick access-time. Should such a storage exceed the capacity to be expected of electronic computer-like machines of the near future, other approaches exist that will trade smaller lists for more operations. But this is already beyond my present topic.

Discussion

Oswald: Bar-Hillel's estimate on the time it would take to prepare an Operational Grammar seems to me unduly optimistic. I quit my own work on the German Syntax just because after the completion of the first steps the problems multiplied to such a degree that I was unable to handle them.

Reifler: I think that a good linguist with experience in this type of work could do it rather rapidly.

Bull: You are both wrong. In my work on this problem in Spanish, I was able, in eight years work and a cost in labor of about $25,000, to deal with approximately 60% of the language. With an appropriate staff, I would say it would take four more years to complete the job.

There are some problems in Spanish Syntax which I would like to solve, but the trouble is that you could not find an example for such a construction in less than 845 pages, on the average. And I need 500 examples of this construction before I can determine the operational pattern.

Locke: Could we not start with A mechanization of, say, 78% of the syntax? It is amazing how good 75% can be when you get this much.

Bar-Hillel: I would like to start with 90%. Then I would let the machine run for six months and on the basis of the experience gathered thereby increase the percentage of the syntax mastered by the machine.

Wiesner: I would like to get more illumination about the meaning of this percentage business. You know, when I was working with Bell Telephone, I was shocked when I discovered that a coefficient of 70% in certain articulation tests was regarded as satisfactory. But I wonder whether we have something of this sort here. What does it mean that the translation machine performs satisfactorily to 75%. Does it mean that in 25% of the time we will get something that does not make any sense at all or does it mean that in all cases we will get something that is 25% inaccurate and vague?

Reifler: With regard to another point in Bar-Hillel's talk. I would like to ask whether it is technically possible to have the machine follow up eventual multiple syntactic resolutions simultaneously. This would certainly mean a great saving in time.

Bar-Hillel: In principle, such a procedure is certainly possible, but it means that we would have to build two or more machines working in synchronisation. And it is questionable whether the gain in time in worth this additional expenditure. A million operations at, say, 15 microseconds per operation would require only 15 seconds; two machines working simultaneously could reduce this time on the average to 7½ seconds. This does not seem worthwhile.