[*Eighth Annual Round Table Meeting on Linguistics and Language Studies*, Georgetown University, 1957]

The Requirements Of Lexical Storage

Gilbert W. King
International Telemeter Corporation
Los Angeles 25, California

Lexical Search

In recent studies of Machine Translation a good deal of attention has been paid to translation, but very little to machine. There seems to be a feeling the machine will be more or less like existing computers. Such an assumption must be taken with caution.

There are two ways to carry out computations on a machine. One is to construct the required result by algorithms; for example, the quantity *sin x* can be calculated by a repetitive formula equivalent to a power series. The other is to rely heavily on table look-up. In present-day computers the latter method is almost extinct, and in Mechanical Translation we must strive as much as possible toward algorithmic methods.

Inasmuch as it seems impossible to construct the meaning of a word from its spelling or phonemes, except in the few cases of onomatopoeia, Mechanical Translation must always rely heavily on table look-up rather than algorithmic methods. Furthermore, a word not only has its dictionary meaning, but also the adhesion of a great deal of psychological and unexpressed descriptive material. A “sack” and a “coffin2 are both “containers”, but it would take a paragraph to modify the word “container” to make it mean either “sack” or “coffin”. Thus in order for the machine to choose the most appropriate word of this category, we must store away additional material with each word to aid, to the degree of sophistication required, in the ultimate selection.

So although we expect to look up meanings associated with words, we do not wish to have an automatic dictionary, but to de-emphasize this approach, and try to introduce as many algorithmic techniques making use of context as possible. Thus we should consider “lexical search” rather than “dictionary look-up”.

Magnitude of the Search Problem

The extent of the lexical search is determined not only by the theory, but by practical limitations. We now know that Mechanical Translation is possible, probably to as high a degree of refinement as we wish, so what are our objectives now?

Are we to pursue Mechanical Translation as an academic stunt? Do we expect to turn out useful translations, but presume they will always be crude and inelegant? Are we to provide a means to translate a specific field such as science or technology, or all types of literature?

The first is not enough, the last beyond our capabilities now. But the second is possible in 1958. In fact our objective should be to translate scientific or technical material in accurate readable form, with one proviso. Such an effort would be of great value to the nation, only if it can be done as fast as foreign presses print the material. The problem of lexical search is what is known in the computer field as a “real-time problem”.

No hardware yet exists to carry out Mechanical Translation in real time. The current output of the leading nations is of the order of 3 x 10^6 pages per year, or 10^9 words per year. In the next year or two we may expect text readers to be developed which will be able to read printed material at the rate of 1000 characters/sec. With 10^7 sec in a working year and 6 characters/word, this amounts to 1.5 x 10^9 words/year, of the order of magnitude of the rate of publication. Thus we can expect the rate of input to the machine to be adequate.

Storage for Lexical Search

The corresponding rate at which the lexical search must be carried on is 10^9 words/year or 100 words/sec. Thus the first requirement on this memory unit of the machine (which we shall call Store I) is that it must have 10 millisec random access time to every entry. It will take one-fifth of a second to look up all the words in an average sentence.

The size of the store will depend on the number of words in a language, and the amount of lexical material to be associated with each word in an entry. There are some 6 x 10^4 words in a dictionary. However, at the present state of translation theory we can hardly afford to neglect the clues offered by inflexional forms, so the total number of source words which must be in the store will be more like 10^6. At present we average about 250 bits (6 bits to define a character) in an entry, and more sophisticated translators will require about 10^3 bits. Thus the second requirement of the store is that its capacity must ultimately be about 10^9 bits.

There is a third design parameter of the store which must be established to make the translating system efficient. Access to an entry has been established at 10 milliseconds; the size of the entry at 10^3 bits. This material must be read out in a reasonable fraction, say 10%, of the access time. Thus the third requirement of the store is that its read-out rate must be 10^6 bits/sec.

Storage for Logical Processing

The problem of lexical storage involves more than the mere storage and access to lexical material. A good translation also involves the interrelation of the lexical material found on the basis of syntactics and semantics. The first disgorgement of the store is only raw material, on which a logical unit of the machine has to work. (Here "logic" means that mathematical or symbolic logic which can actually be done with a computer. Some "logical" operations are purely housekeeping details of the mechanical operations of the computer.) Rough estimates based on current theories of Mechanical Translation would indicate that some 10^4 logical operations may be required per sentence to straighten out the disgorged material into a good translation. Even if only a fraction of these operations were requested for further look-up (as many theories demand), the restriction of producing output as fast as material is fed in makes it imperative that no further look-ups in the large store are permitted during logical processing.

This means that the disgorged material on the first look-up (from Store I) should be necessary and sufficient for analysis of the sentence (or paragraph). In other words, the output of the first look-up operation creates a "microglossary" sufficient for the analysis of the sentence. This selection from Store I should be dumped in a fast memory (called Store II) for logical processing. With 20 words per sentence on the average, 10^3 bits output/word the requirement on capacity for the intermediate memory is of the order of 10^5 bits (100 thousand-bit computer "words").

We have seen that the rate of flow, from source through input equipment and in table look-up in Store I, are all well matched at 0.2 sec per 20-word sentence. High-speed memories of 10^5 bits capacity are currently available with a 10 microsecond random access. Hence $0.2 + 10^{-5}$ or 2×10^4 logical operations (computer-type) may be made with the microglossary. Since some 20 computer-type housekeeping operations are normally required for one purely logical operation (e.g. a comparison of endings), about 10^3 of the latter are permitted per sentence. Of these perhaps 10^2 may be further table look-ups (in the fast memory). This facility seems adequate for current Mechanical Translation theories.

Nature of the Logical Processing

This scheme of setting up a microglossary for each sentence imposes not only the above physical requirements on the intermediate memory, but also begins to define the logical elements necessary in the entries.

At this point in the machine it is actually unnecessary, and is in fact premature, to have any translation into the target language.

We may cover most of the theoretical approaches by defining the contents of the entry in Store I as clues. That is, given a sequence of words in the source material

$$S_lS_2....S_i.....$$

the first operation is to look up in Store I lexical information concerning the words S_i (or word sequences S_i, S_{i+1} . . . S_{i+k}). The output will be a sequence of terms

$$S_1A_1B_1, S_2A_2B_2, ... S_iA_iB_i ....$$

where A_i refers to characters (possibly binary) giving syntactical information (such as the part of speech), and B_i refers to characters giving semantic information, e.g. “this is a word from physics”; but not the translation.

The sequences ($S_iA_iB_i$) form an expanded sentence, and form the microglossary in the intermediate fast memory of the logic portion of the computer. Here the sequences A_i, B_i are examined and a new set of characters C_i are constructed and assigned to each S_i. Note that the i’th C_i, assigned to S_i, is in fact a function of all the preceding and succeeding A_i’s and B_i’s (called the “local” or “minor” context). The determiners (A’s and B’s) for the C’s may not only be in the sentence, but possibly (especially for pronouns) lie in previous sentences, or even of the title (field), called the “Major Context”.

These logical operations will consist of two groups. The first will be a syntactical analysis of the sequence A_1, A_2 . . . A_i (without the S_i’s or B_i’s). This is like a schoolboy’s diagramming of the sentence, which finds the relations between words. According to the Cambridge Language Research Group this analysis can be made by algebraic lattice theory, which is highly algorithmic.

To give a very elementary example from French let all nouns have an A=a, all verbs an $A=ß$ and the word S=le have an A=$a+ß$. Here the plus symbol is the logical “or” operation. There will be as many terms in A as there are multiple meanings for the S. Then the syntactical analysis of “le” followed by a noun would involve the Boolean multiplication, which is easy to mechanize,

$$(a+ß)(a) = a$$

The result, a, would constitute a character of the C for “le”, so that the output SC for “le” would be le$a$. The augmented word le$a$ has a unique meaning “the”. Note the actual meanings of the augmented words S_iC_i are not yet at hand, and are to be found by a third operation in the machine, to be described below.

In the case of “le” followed by a verb, the multiplication is

$$(a+ß)(ß) = ß$$

and the output SC for “le” would now be le$ß$. The augmented word “le” has the unique meaning “it”. (The other meaning “him” would be assigned to “le$a$”, derived from other A’s.)

A more complicated example would be the phrase

“. . . penetrée d’abord de ...”

in which the logical operations on the A’s for the four words (d’abord not being treated here as an idiom) should show that “de” not “d’” is modified by the “penetrée” (and is ultimately to be translated as “by” not “of”).

According to the MIT group these operations will require another series of table look-ups. The storage involved probably does not require high capacity, but will require fast access, and be similar to Store II.

When the permissible connections between words has been established by purely syntactical analysis, by means of the A’s, a second series of operation, involving the B’s, is carried out. Consider the following three elementary examples in French.

1) . . . le livre est à lui . . .
2) . . . il est pour travailler . . .
3) ... pour . . .

In sentence 1) “est” has a specific meaning “belongs”, the clue for this selection being “à”, which has its normal meaning “to”. In sentence 2) “est” has its most probable meaning “is”, but “pour” is to mean “about to” here. In sentence 3) “pour” is to have its most probable meaning “for”. We shall not complicate matters by giving sentences where “à” is controlled by other words giving it meanings other than “to”, but remember this in the formulation. To handle the multiple meanings for the three words est, à and pour, whose clues are specific words elsewhere in the sentence, rather than purely syntactical, we assign to these words B’s which are logical sums of characters a, b, c . . ., one for each possible meaning. Thus for

S = est	B = a + b + . . .
S = à	B = b + c + . . .
S = pour	B = a + d + . . .

(Subscripts to the S’s, used previously, giving the position of the word are omitted.)

Then in the first sentence we would have

$$(a + b + . .)(b + c + . .) = b$$

to get estb and àb.

For the second

$$(a + b + . .) (a + d + . .) = a$$

to get esta and poura. For the third we would get only poura, and for sentences where “à” has other meanings àc etc. The corresponding augmented words have the following unique meanings

esta = is ~
estb = belongs

àb = to
àc = of
poura = about to
pourd = for

The output of the logical unit is then a sequence of

$$S_iC_l, S_2C_2... S_iC_i...$$

The point here is that we are no longer concerned with raw words S_i of the source language, but augmented words S_iC_i, and these augmented words, if our method of construction of the C_i's is adequate, have a unique meaning.

At this point in the machine we should have then solved the multiple-meaning problem with the aid of the syntactical and semantic context.

Output Store

We now come to the final stage of the machine, which again is a memory look-up operation. We enter with the individual augmented words S_iC_i and find a single target equivalent T_i. (Note S_iC_i may stand for a string of words S_k . . . S_m, from which some S_kC_k have no target equivalent.)

The statement that the machine has a second look-up in a large store for each word does not violate our precept that time does not permit more than one look-up, because this operation is on another store, and can be done in the interval when the preparatory look-up for the next sentence is going on. (It is reasonable to suppose the intermediate Store II is flexible enough to be accepting $S_iA_iB_i$ from the first memory for the following sentence while simultaneously supplying S_iC_i for look-up in the last memory for the sentence in hand.)

Nevertheless the speed of the last large memory (Store III) must be such as not to delay the overall flow of information through the system.

Since the logical operations have only made a one-to-one correspondence between the $S_iA_iB_i$ and S_iC_i, the number of look-ups for the sentence remains the same. Thus the requirement on the last store in regard to access time is the same as for Store I.

In simple Mechanical Translation theories, on the average there are 3 multiple meanings for each source language word, the number of entries in Store III will be three times that of Store I. Further the length of the address, S_iC_i, will be about twice the length of the address S_i used in Store I. On the other hand the information sought is only a simple target equivalent, averaging 6 characters, or less than 50 bits. The length of an entry will thus be about 150 bits. The total capacity of Store III will be about one-third that of Store I. Nevertheless, in view of the rudimentary state of the theory, for the following reasons one should consider Store III as having essentially the same capacity as Store I.

The Thesaurus

It seems that a more advanced theory of Mechanical Translation, or more accurately, of mechanically understanding the written word, could be developed along these lines. The semantic information B_i associated with each input word S_i in the first lexical search, could be elaborated in great detail; so much so that the output of the logical unit could dispense with the symbols, S_i, of the source words, and be merely a string of C_i's,

$$C_1C_2...C_i...$$

This presupposes that the B_i's, and the analysis of relationships by means of the A_i's, are sufficiently detailed that the sequence of C_i's has retained all the content and relationships the whole idea, in some coded form related to symbolic logic. In this event Store III would be a kind of thesaurus, for which the input is a sequence of symbols, C_i, associating in a Boolean function a large number of ideas and relations which must be stated in the output, as determined from the initial contextual analysis; and for it we wish the machine to choose the most appropriate word. This word is not necessarily the one we would find in a dictionary, nor is it a synonym, but a particularly cogent word for the idea in the particular context. In passing we remark that the C_i's themselves constitute a language analysis to symbolic logic or the proposed "ruly English", but are unsatisfactory output in themselves as they do not convey the richness and desirable ambiguity (after Empson) which makes ordinary languages sophisticated means of communication. In short the thesaurus reattaches to the primitive C_i's the psychological content and background description that makes languages.

In order to point out that the effort spent on both the theory and hardware for Mechanical Translation is of value not only in itself, but for the larger problem of information retrieval, we may point out that in the above system the output T_i from Store III may indeed be the same language as the input S, so that the machine translates English into better English. Or T_i may be the more primitive English used by librarians and indexers, so that the system could be used for classifying, indexing and abstracting.

Incomplete Matching

There is an important point in imagining the construction of local context and introduction of the thesaurus in contrast with a dictionary.

Inasmuch as the C_i's are determined from the local context, which, if the material is worth translating, should have some novel combinations of ideas, we cannot expect all possible C_i's to be listed with an S in Store III. That is we do not necessarily have unique addresses to the entries of Store III. Hence we must arrange to locate not necessarily a specific C_i, but a best match. There are various ways of defining “best”; one is, recognizing C_i to be essentially a Boolean function, to find a C_i which dominates C_i in the sense of lattice theory, i.e.

$$C'_i \geq C_i \supset T_i$$

A system such as this will have to be introduced even in simpler Mechanical Translation schemes, to handle typographical errors and grammatical errors on the part of the original author.

Summary

The Mechanical Translation system consists then of three parts, first a high capacity millisecond-access store of lexical information concerning the source language; second, a low-capacity microsecond-access store for logical processing of lexical information into augmented words for selection; and third, another high-capacity millisecond-access store of thesaural information concerning the target language. The whole system must operate in real time.