[From: *Computers and Automation* 2 (4), 1953]

MECHANICAL TRANSLATION

by Andrew D. Booth
Birkbeck College, London, England

During the summer of 1947 I first suggested that a digital computer having adequate memory facilities could perform the operations necessary to translate a text written in a foreign language (F L) into the desired language or target language (T L). There was, and is, no particular difficulty in doing this, as I hope to show in the present article; but I make no claim that a literary quality in the result of the translation is to be hoped for.

The original proposals covered only the making of a straightforward dictionary translation from the foreign language to the target language. It is convenient to start by seeing how this simple objective may be achieved on a machine whose primary purpose is the manipulation of numbers. It is necessary to assume only the most rudimentary machine functions in order to perform mechanical translation (M T):

a) The machine has a large memory.

b) The input typewriter sends data, either direct to the memory, or to a register provided with subtraction facilities, the accumulator register.

c) The machine contains a conditional transfer order which enables the machine to select between alternative courses of action according to the sign of the number held in the accumulator register.

d) The contents of the accumulator can be typed at the output.

The reader familiar with modern automatic digital computers will see that all of the above functions are present in all such computers existing, with the exception in many cases of the large memory.

How shall we represent the foreign language text in digital form? A normal teletype machine is so constructed that the depression of any key, for example that corresponding to letter A, causes the emission of a binary coded digit pattern which has a one-one correspondence to the desired character. Thus:
A becomes 00011; B becomes 11001; C becomes 01110;...; and Z becomes 10001.
It follows that, if the keys corresponding to the letters of the foreign word are depressed, in sequence, a digital pattern will be generated which uniquely represents that word. If this pattern is regarded as a number, a dictionary translation of the foreign word can be obtained by storing the translation in that memory location which has the same number as the code of the foreign word. As an example, the Latin word <u>et</u> is coded 10000, 00001, which as a binary number is equal to 512 plus 1 or 513 and would identify memory location 513. Then in that memory location 513 we would store the translation: 00100(d), 01100(n), 00011(a) corresponding to "and". The reader interested in details will notice that it is assumed that digits are shifted into the machine register, starting from the least significant (right shift), and that the inversion of order (d,n, a) is necessary for the output type to appear in the normal sequence.

It is at once obvious that this simple scheme is quite impracticable, since even in the example given, it will be seen that 1024 locations are required to deal even with the two letter words of the foreign language. For words of maximum length say 10 letters, 2^{50} locations would be needed. This would exceed even the most sanguine hopes of modern machine designers. In any case no known foreign language has anything approaching 10^{15} (2^{50}) words, so that almost all of the memory would be empty.

The difficulty is easily overcome, however. Suppose that each location (in sequence) in the memory contains a "dictionary" word (D W) having the following composition: the F L word (10 letters say) and the T L translation (40 letters say). Assume that the D W's are stored in ascending order of magnitude. Then if the F L word is subtracted from each of the D W's in turn, the result will be negative until the required entry is reached and positive thereafter. It follows that, if the conditional transfer is used to break off the sequence of subtractions at the first positive result, the remainder in the accumulator at this point will represent the target language translation. The latter may now be printed at the output.

A second obvious point is that the length of the required words (250 binary digits or bits in the above example) is considerable. Existing computing machines fall short of dealing with this by a factor of five or greater. They may, however, easily be programmed to use multiple length words so that this is not an essential difficulty.

If the actual F L word is not contained in the memory, the nearest equivalent will be generated by the above process. Furthermore, since the D W, F L entry will be numerically somewhat larger than the text F L word, the output operation will generate certain nonsensical characters before the T L translation. This will indicate to the reader that an untranslatable word is present.

The preceding simple scheme is much limited by the available memory in existing (and near future) machines. But in 1948 R. H. Richens suggested to me a modification which makes mechanical translation a really practicable operation. Richens pointed out that, with certain limitations, an adequate or passable translation of a foreign language text would result from the following operation:

a) The memory contains a stem (or root) dictionary and an ending dictionary.

b) The stem dictionary consists of a relatively few entries of general semantic utility plus a vocabulary specific to the subject of the translation.

(The latter has since been called, by V. Oswald, Micro-semantics).

The method of operation is simple. First the F L word is subtracted in turn from the entries in the stem dictionary. In this way, the longest possible stem entry is found. At this point the stem translation and suitable grammatical notes are typed out. The stem is now removed from the F L word, and the remainder is compared with the entries of the ending dictionary. When coincidence is attained again, the relevant syntactic information, contained with the ending entry, is typed out.

Richens has shown that the same method can be applied to multiple words of the type encountered in, for instance, German.

As an example of this procedure consider the translation of the Latin word amo. This would proceed as follows:

Stem: Trial 1: a, alas
Trial 2: am, love (v) (v for "verb")

Ending: Trial 1: o, (1.s.p.) (for "1st person singular, present tense")

The total output would be: love (v)(1.s.p.)

Certain difficulties arise, as in the example desideremus given by Richens. Here two possible translations exist: (1) desider, desire; emus (1.p.s.a.); or (2) desid, be idle; eremus (1.p.i.s.a.). Resolution could be attained by storing the word itself, together with both translations.

Again, certain words, or parts of words, are sometimes without significance, for example the t in the French a-t-il. In this case, to avoid confusing the operator, the machine probably would have to put out some encouraging symbol, such as "N" for no significance.

It has been suggested, by Prof Erwin Reifler of the Univ. of Seattle, Washington, that semantic ambiguities could be considerably eliminated by the use of a person called a "pre-editor" who could be a native in the F L but would not necessarily know the T L at all. The duty of the pre-editor would be to replace all ambiguous words by non-ambiguous equivalents.

The foregoing brief account of mechanical translation is naturally incomplete in many respects. The act of coding a given example for a particular computer involves many points which it has been impossible to cover in a short article. This is particularly true of the stem-ending dictionaries, whose use requires a high degree of sophistication in the program if a good working speed is to be attained.

Some of these problems however have been actually examined on our computer APEXC at Birkbeck College, London, and the reader may be interested in the following statistics:

Time taken to translate a 1000 word message by a skilled bilingual human being	1 hr.
Time of mechanical translation using the above technique, on standard punched card equipment	1 hr. 54 mins.
Time of mechanical translation on APEXC using teletype output	2 hrs. 15 rains.
Time of mechanical translation on APEXC with tabulator output	30 mins.

It does not appear likely that with existing input-output equipment any much greater speed is possible. The translations, produced by the above methods are of course inelegant, but are easily understood by a person expert in the subject of the paper. Neither the present author nor Richens envisage the literary use of mechanical translation in the near future or even foreseeable future; but within its limitations, the method should be of great use to students and institutions confronted with the mass of published material in foreign languages which is currently appearing.