[*Information processing: proceedings of the International Conference on Information Processing*, Unesco, Paris 1959] 163

A reduction method for non-arithmetic data and its application to thesauric translation

By A.F.Parker-Rhodes and R. M. Needham, Cambridge Language Research Unit,
Cambridge (UK)

Work on mechanical translation has shown that it is possible to represent the semantic fields of words by means of a thesaurus, and to use the thesaurus for the operations involved in translation, provided that this information can be made available to a machine in a form which is neither too bulky nor too complicated for economic use. It was thus necessary to have available in as small a compass as possible a large bulk of data, consisting of items each of initially 1000 bits, to be operated on for translation purposes by Boolean operations.

Methods have been developed to reduce as far as possible the number of bits required for each item, consistent with simplicity in carrying out the Boolean operations, and practicability of performing the reduction. The methods described are generally applicable to bodies of data which are used for logical purposes (of which matching is the simplest case), and are interesting because they show an unexpected possibility of economy without any loss of information in the storage of such data. They are related to the problem of finding the optimal Boolean encoding of a given partially-ordered set or of a lattice, a problem which occurs in its simplest form in the allocation of function-digits in a computer.

The relation of the procedure to methods of code-compression which involve limited but predictable loss of information, is discussed in connexion with the treatment of new accessions to encoded data. The computing procedures involved are illustrated by reference to the case of the cross-reference dictionary of Roget's Thesaurus.

1. Introduction

Among the essential requirements for a practicable machine translation procedure is some way of presenting what we call the "meaning" of a word such that calculations can be made upon meanings, just as in arithmetic we make calculations upon numbers. We shall not expect that the words of a good translation of a given text will bear a one-to-one correspondence with those of the original; and even when such a correspondence can be established (as is often the case in simple sentences or in related languages) corresponding words will agree in meaning only in the given text, and will only rarely correspond in all possible contexts. Even if we had some way of writing down the meaning of any word, and of recording these symbols in a dictionary, it would not be sufficient to look up the meaning of each word in the source text in the dictionary, and then look up the dictionary of the target language in reverse, so as to locate that target-language word which had the same meaning. Obviously, no one word of the target language would be expected to have exactly the same meaning as the source word. But it equally would not do to look simply for the "nearest" target-language word; for we are not interested in the total range of meanings of a word, which is what the entries in such a dictionary would give us, but only in the particular meaning it has in the given context. This implies some way of discovering the meaning that a particular word has in a particular context, given the total range of meanings of all the words in the source text, which text tells us all we know about this context. This is as much as to say, that we require an algorithm in which a set of word-meanings are manipulated so as to yield another word-meaning, which in general will be different from any of those we started with.

One's first thought, in such a situation, is to enquire whether the required algorithm could be reduced to looking up a table. Could we not list all the meanings of each word, in some notation, and provide a table of correspondence whereby in any given context the right one could be picked out? Many arguments show that this would be impracticable. In the first place, it is impossible to present a complete list of the meanings of a word, either because new usages are always being created and are in practice very frequently encountered, or else because, as with many prepositions, the list would be indefinitely long. Apart from this, any general table of correspondences would be prohibitively large (the number of entries would be of the order of the square of the number of words in the language) while special tables, differing for different words, would result in the dictionary entries being impossible long. Since, therefore, a referential algorithm will not serve us, we must make provision for an operational one.

Two requirements have been pointed out above which such an algorithm must fulfill. It must be possible to construct a metric space in which the word-meanings in a dictionary can be located, so that we can give a meaning to looking for the "nearest" word to a given prescription, And secondly, it must be possible to calculate the meaning that a word in a source text carries, in a given context, from the meanings of the word itself and not too many of its neighbours in the text. The first requirement says in other words that we must be able to find a word, within a minimum margin of tolerance in the target language, when we are given what is required to mean. The second one says that we must be able to find what each word is supposed to mean, if possible with no margin of uncertainty.

Now there exists a class of linguistic compilations which have the aim of presenting the words of a language in such a manner that given an idea, more or less clearly defined, of what he wants a word to mean, the user can find words of the language which will express the required meaning. Such a work is called a thesaurus. A thesaurus evidently fulfills, or is intended to fulfill, the first of our requirements, The C.L.R.U. therefore, some while ago, undertook experiments [1—4] to discover whether an actual thesaurus (the English one of Roget) could be used specifically for translation by a programmable procedure. Although this can, up to a point, be achieved, it was found that Roget's thesaurus was unsatisfactory for this purpose in the following respects:

1) the cross-reference dictionary is grossly incomplete;
2) many words ought to be included under many more heads than they are in the present thesaurus; especially in the case of closely related heads;
3) the thesaurus is very defective in lists of technical terms, and classes of unambiguous terms such as names of plants etc.

This is not the place to discuss in detail the linguistic and philosophical aspects of the thesaurus, and papers on these topics are at present in course of production; only its mathematical features will be considered here. In what follows we are obliged for want of space to assume a knowledge of the elementary terminology of lattice theory; the application of this theory in this field is no longer a novelty, and to explain our ideas without its aid would be very lengthy.

2. Representation of the thesaurus

It became clear, from our analysis of the thesaurus, how it could be represented as a metric space, and this provided the answer to the second requirement, that the word-meanings should be capable of algorithmic manipulation. Examination of the thesaurus showed us that what was recorded for each word was in effect a list of its possible uses; not, of course, one by one, for there would be an infinity of them, but by classes. All the words treated are listed in the thesaurus under one or more of a thousand heads. Each of these heads is, in effect, a class of word-uses, such that, ideally, any given occurrence of any given word can be allocated unambiguously to one and only one class. I say ideally, for in fact this is not always true: this is one of the deficiencies of Roget's thesaurus which will have to be corrected before it can be used for a fully mechanized translation procedure. Nevertheless it seems clear that such an arrangement is what a thesaurus aims at. If it were fully realized, the thesaurus would consist of (a) a classification of word uses, on some principle which we must discuss shortly, and (b) a definition of each word of the language in terms of those classes to which its possible uses belong. These definitions clearly constitute a partially ordered set, under the ordering relation that A includes B if every class of uses recorded for B is recorded also for A. This partially ordered set is presented as embedded in the Boolean lattice B^n, where n is the number of heads in the thesaurus and, as we shall see, could be embedded in a smaller Boolean lattice if desired. Any partially ordered set is a metric space, and a possible metric is provided by the least number of links in the Boolean envelope required to pass from one element of the set to another. This may not be the most appropriate metric for our purpose, but it has the merit of being very easily calculated from the raw data provided by the thesaurus.

Representation of the thesaurus as a partially ordered set embedded in a Boolean lattice also provides us with a simple set of algorithms with which to manipulate word meanings. For as defined by the thesaurus, each word is represented by a list of heads; if there are n heads altogether, any such list can be regarded as an element of B^n. Once word-meanings are interpreted in this sense, we have a great variety of operations definable in B^n wherewith to operate on sets of meanings in order to arrive at an answer to the question "what does this word mean in this context." The use of these operations to define the algorithm we require implies that any element of the Boolean algebra B^n could be the meaning of a word in some language, and in particular that any such element could be a prescription, the "nearest" approximation to which represented by a word of the target language could be reliably taken as an appropriate translation for the source word which yields it. This is so only if the classification of the word-uses on which the thesaurus is based is suitable for the purpose (for instance, an alphabetic classification obviously would not do). That in fact Roget's thesaurus uses a fairly suitable classification is proved by the fact that it can be used, however haltingly, to produce actual translations. One necessary property of such a classification is that it could be used to classify the uses of the words of any second language, actual or conceivable.

There is, however, an evident difficulty of a practical nature. The algebra we propose to use is B^n, and, as we have mentioned, in the case of Roget's thesaurus n is 1000. It does not seem realistic to expect that this is merely a whim of Roget's, for independant considerations of a diverse character point to this order of magnitude as likely to be encountered in any usable thesaurus; it has therefore probably some empirical justification. But there are serious objections to attempting to perform operations mechanically in this algebra, on the grounds of the size and cost of the equipment required if a commercially adequate speed is to be attained. For each element of B^{1000} is represented by a computer-word of 1000 bits, whereas the word-length of actual computers is normally well under 100 bits. Moreover, even if this difficulty were overcome, it would remain the case that even the most ample dictionaries in any language contain at most 2^{20} words or so, which occupy only a negligible fraction of the 2^{1000} elements at our disposal; consequently this algebra exploits the capacity of any computer with astronomical inefficiency. We are therefore impelled to ascertain whether the given partially-ordered set, represented by an actual thesaurus such as Roget's, could not be embedded in a substantially smaller Boolean lattice than B^{1000}. If it could be, the method we envisage could lead directly to a practicable translation programme.

We should however first consider the possibility that such a re-encoding of the thesaurus is not really necessary. It might be the case that, even without deserting B^{1000} for a smaller algebra, we could find some practicable algorithm which could be proved equivalent to the use of B^{1000} but which could be quickly performed on an ordinary computer.

Two means of achieving a better encoding of the items than 1000 bit entries immediately suggest themselves. First we may give as the specification of each item, simply a string of numbers which correspond to the digits in which there is a 1 in the complete encoding. Since there are 1000 bits, we shall, if the mean number of ones is r, require on average 10r bits per entry. This method is exceedingly inefficient, both because the variable length of the items is very wasteful of storage space, and because the time taken to perform Boolean operations on items represented like that is rather great. It would essentially be necessary to perform merging or collating operations on strings of numbers and although a fast machine can do this very quickly it is still quite a complex operation by comparison with the simple Boolean operation on Boolean elements.

Secondly, we might use the methods of superimposed coding to reduce the length of the entries. This proves to be undesirable for a different reason which appears when

the probabilities of errors are examined. In our application we have not only to retrieve items corresponding to a certain specification, but we have to perform Boolean operations of some complexity on the specification first. Furthermore the numbers of heads allocated to words are widely variable, that is to say the number of ones in the unabbreviated 1000 bit entries is very variable. It is easy, though somewhat tedious, to show that both these differences in purpose reduce the efficiency of superimposed codes very much. The algebra is not given here but its conclusions are as follows:

1) It can be shown that the result of a Boolean operation on n operands each of m bits, any one of which has a probability p of being in error (i.e. of being 1 where it should have been 0 or vice versa), has a probability mnp of giving a wrong result. In the translation procedure we envisage the number of operands used in calculating a single translation prescription may be as high as ten, and m will be at least 40. A certain proportion, perhaps about a third, of the prescriptions will permanently affect one or more context indicators which will serve as operands in computing further prescriptions up to say the end of each current paragraph. This last point introduces a cumulative element into the process, so that any errors in any of the prescriptions will have a chance of affecting not one but many words in the output. Altogether, the chance of some error occurring within a paragraph of 100 words may be as much as 10,000 p. Thus even if p is as low as 10^{-6}, which is better than the achievement of any code-compression at present in use, we are left with a possible chance of error of 1%, which is about as much as we should be prepared to tolerate.
2) the probability of error increases rapidly if the number of heads (ones) in the specification has not only a high mean but also a high variance.

In our case the number of operands may be very large and the second condition is by no means satisfied. We therefore had to abandon the use of superimposed coding on any random basis. After this prolegomenon, which is intended to show that a new method was indeed required, we may describe the procedures that were adopted.

3. Procedure adopted

3.1 Any set of elements of B^{1000} will constitute a partially ordered set under the Boolean inclusion relation. Furthermore the partially ordered set may be used to define a lattice uniquely by inserting elements of B^{1000} wherever necessary to satisfy the lattice axioms. This process may be made unique, for the ways in which the lattice axioms are contravened are

1) absence of 0 & 1 elements, which are simply inserted
2) Presence of a pair of elements without a unique join, in which case the Boolean join (meet) can be inserted. Notice that we have here an important consequence of the fact that the partial-ordering properties of the set is already represented by a Boolean symbolism, for we know that there is always a Boolean element there to be used.

There will now correspond to the original Boolean operations, operations of taking meets and joins in the lattice derived from the partially ordered set; we therefore wish to achieve the best encoding of the elements in which these operations may be performed.

If we assume that the encoding, when found, will be a Boolean one, we may say that we are trying to find the degree of the given lattice, that is to say the degree of the least Boolean lattice in which it can be embedded. In other words we are trying to find the least number of bits in which symbols can be set up for the elements such that

1) no two elements have the same symbol;
2) no element appears to include any other element that it does not in fact include.

The authors have worked out theoretically complete methods for finding the degree of a lattice and effecting a reencoding of its elements in terms of the appropriate number of bits per symbol. These methods are however, in this context, largely of theoretical interest as the lattices which in fact occur in linguistics are very large.

It is however possible to predict, by statistical means from a sample, what the theoretical minimum may be expected to be in any particular case, and to perform a reduction by means of an iterative procedure, the effectiveness of which may be judged by comparison with the statistically predicted result.

3.2 *Statistical estimate*

The principle of the method is to construct a lattice in such a way that its degree is known beforehand, and some at least of its structural properties agree with those of the given lattice. The properties chosen have to be (a) reasonably easily ascertainable for the given lattice, (b) likely to be sufficiently critical for accidental agreement to be unlikely, and (c) calculable for the constructed lattice.

The structural property which we have chosen is the frequency distribution of elements which include increasing numbers of minimals of the lattice. This distribution can be found for Roget's Thesaurus as follows. A sample of the words in the cross-reference dictionary of the Thesaurus is taken, and for each word is recorded the heads under which it is said to occur and (this is the only part of the information which we actually need) the number of heads for each word. This information is punched on Hollerith cards, and it is then simple to ascertain the number of sets of words (representing elements) containing any given number of heads by using a sorter. The heads themselves then correspond to the minimals of the lattice.

3.3 *Constructing a container of known degree*

Take the lattice B^n whose degree is n. Choose therein a level and delete every element below 1 and above the bottom element of B^n. Delete moreover all but a fraction p of the elements in level 1. The system remaining can easily be shown to be still a lattice, and its degree is not more than n (if 1 is not above the equator of B^n the degree is exactly n). The remaining elements of level 1 will be the minimals of the constructed lattice. Let us denote the number of elements of this lattice L (n, 1, p) which include i minimals, by f (i).

If we choose at random k elements from the level 1 of B^n, the chance that exactly i of these k will be also elements of L is.

$$\binom{k}{i} p^i (1-p)^{k-i}$$

The number of elements of level l which are included in any one element of level 1 + j of B^n is $\binom{1+j}{1}$

Therefore the number of elements of level 1 +j of L which include exactly i minimals of L is

$$\binom{\binom{1+j}{1}}{i} \binom{n}{1+j} (1-p)^{\binom{1+j}{j}} \left(\frac{p}{1-p}\right)^i$$

Therefore the total value of f (i) is given by the sum of the above quantities for all j from 1 to n — 1. This quantity will depend on n, 1, and p. These, however, are not independent. Firstly the number of minimals, equal to the number of heads, which is actually 1000, must be equal to p (n1). Secondly, the formula gives the distribution function as a binomial series. This is difficult to evaluate but

since the initial element of the general term is invariably integral and at least one, we may instead evaluate the geometric series

$$\sum \binom{n}{1+j} (1-p)^{\binom{i+j}{i}} \left(\frac{p}{1-p}\right)^{i}$$

to get a bound to the result. From this a limiting value of p can be ascertained from the statistics of the given lattice. The frequency distribution, as found from a sample of 712 words out of a total of about 25,000, proved to be of the negative-binomial class, not far from the logarithmic series familiar in biology. It is easy to find the smallest value of the base b for which a geometric series will nowhere fall below the empirical distribution. The minimum value of p satisfying this criterion was 0.461, whence b = p/(1 — p) = 0.855. Then, for each n there is one and only one level 1 minimally consonant with the given number of minimals, 1,000.

The procedure then consists in calculating the distribution of f(i) for various values of n, and finding the smallest n which gives a distribution, no term of which is less than the corresponding term for the given lattice. The values of the latter are taken, on the assumption of homogeneity, to be the appropriate multiples of the values of the terms in our 712/25,000 sample. Typical results are as follows (the value of 1 for all these cases is 3):

assumed value of n:	20	30	40	50
calculated value of f(2):	1,962	2,615	9,080	17,410
empirical value of f(2):			4,850	

It appears that the smallest n giving an f(2) not less than 4,850 was 37. The validity of this estimate can be tested in various ways. First, we can calculate from it the frequency distribution which we should have obtained from a sample of that size, which in the given lattice gave us our actual sample of 712 words. This is:

value of i:	1	2	3	4	5	6	7
value of f(i)							
calculated:	165.5	138.3	120.7	103.2	88.3	75.0	61.25
observed:	349	138	84	50	26	16	13

It is clear that there are enough elements in the constructed lattice for the given lattice to be embedded in it, with the exception of the term f(1). But this term consists only of (a) the minimals, and (b) elements of the lattice including only one minimal. The latter need not be distinguished from the minimals they include for linguistic purposes, and these are known to number 1000, which would have yielded us about 30 in our sample. So that the discrepancy does not matter. The 349 entities are words having only one head; these are many times more numerous than the indistinguishable sets of words.

A further check consists in ascertaining the number of elements which include 3 minimals and also at least one element including 2 minimals. This can be ascertained by a similar calculation to that used for the values of f(i). Taking the lattice constructed from B^{37}, it appears that about 0.18 of the elements which include 3 minimals ought to include as well at least one element including 2 minimals. From a (rather small) sample we estimate that the value for Roget is about one fifth. The agreement is close enough to be regarded as satisfactory.

4. Significance of the result

4.1 From the translation viewpoint this result must be regarded with certain reservations, since the thesaurus from which it is obtained is highly imperfect, and the alterations necessary are such as are likely to increase the degree. However, regarded simply as an estimate of the potential reduction in a mass of data, the result is surprising and encouraging. For even regarded as a method of economization for storage and retrieval regardless of Boolean operations in between, it is a much greater reduction than that obtainable from randomized methods of code compression without an excessive degree of confounding.

4.2 *Iterative reduction*

With the knowledge of the expected result, we now proceed to the method for approximating to the minimum encoding.

As explained above it is possible to regard the original data as being encoded by embedding in B^{1000}. B^{1000} is by definition the direct product of 100 factors each of which is B^{10}, which may be obtained by simply taking the 1000 bits of the encoding 10 at a time. Now, however many elements there are in the original data, in any given 10 bits there can be at most 1024 different entries, constituting a partially ordered set of degree at most ten. Since the original embedding is exceedingly "dilute" it is to be expected that most of them will have degree less than ten. In our case it is known that in each 10-bit factor all the minimals will occur. The degree cannot therefore be less than 5, which is the smallest number of bits that will give ten distinct combinations. It is possible using these facts to proceed to reduce the number of bits per symbol by extracting 100 ten-bit factors, re-encoding each economically, and putting them together as a new encoding of the whole data. By adopting this procedure we are at each stage certain that any lattice to be re-encoded as a whole has at most 1024 elements, and a degree of at most ten, the programming of which on an ordinary machine is much easier. After applying this procedure once, we have no reason to suppose that a minimal encoding has been reached. It can be shown that the degree of a lattice is the sum of the degrees of the exponents of its factors if and only if the encoding is already known to be minimal. (The exponent of a factor is the set of elements of it which actually occur. Thus our procedure consists of finding the degrees of successive exponents of B^{10}.)

However the same procedure can be applied again to the new embedding as it was applied to the old one. It will not be futile to do so, for supposing that we obtained the original 100 exponents of B^{10} by taking successive groups of ten bits from the first encoding a similar procedure will yield "second-round" exponents including parts of at least two and possible more of the original exponents. Thus we may proceed iteratively until there is no further reduction in the apparent degree.

The procedure just outlined may be embroidered in various ways:

1) At first it pays, if possible, to take larger factors, say B^{20} instead of B^{10}. This is because the extreme dilution of the original embedding makes it probable that at the first stage the maximum reductions are likely to take place, and because the maximum reduction from ten is to five (see above) but from twenty is to six, which is much larger in proportion. However as the process goes on and the lattice becomes less dilute it may be too slow and complicated to deal with 20 bits at a time.
2) Since the whole procedure is essentially iterative and approximative, it may save time to short-cut the procedure for finding the degree of a set which is already embedded in B^{10}, and to substitute a quicker process which will occasionally give a 'degree' one or two too high. In our program it is initially assumed that the degree will turn out to be five, and characters are allotted to the elements as well as possible on this basis. Higher-level elements are then scanned for confusions and digits added and interchanged to eliminate any that occur. This is substantially more rapid than a program which could cater for all the possible complexities of B^{10}.

Work is at present in progress on machine tests of this procedure using EDSAC II, the data being the same as for the statistical estimate described above. The details of the program would be out of place here; it is sufficient to say that the body of data is stored on magnetic tape, each of the original 1000-bit items being represented as a string of ten-bit numbers giving the positions of the ones in it, and that the exponents are extracted in groups of 25 so that the whole data need only be scanned four times per iteration.

4.3 *New accessions*

The extreme economy of this method has been achieved only at the cost of a loss of flexibility, in comparison with randomized compressed coding. What has been achieved is essentially a compressed coding in which there is no confusion involving two or more items which actually occur. All the confusion takes place with items which are possible in the uncompressed coding but do not in fact occur. If, therefore, it is necessary to add a new item to the system, it is very likely to cause a confusion which can only be removed by adding a bit to the encoding. If this is done, the system will need to be updated by one or two applications of the iterative procedure whenever the number of extra bits used is excessive. For an MT purpose this is probably not too inconvenient, for we shall in any case have to treat our dictionaries as closed for a certain length of time, and update them with collections of new words at intervals. For library retrieval, however, this may be a capital objection to the scheme.

5. References

[1] MASTERMAN, M.: *Linguistic Problems of Mechanical Translation.* Research Report of CLRU, 1957.

[2] MASTERMAN, M., A. F. PARKER RHODES, and M.A.K. HALLIDAY: *Mechanical Translation* Vol.3, no. 1,1956.

[3] MASTERMAN, M.: *The Thesaurus in Syntax and Semantics.* Mechanical Translation Vol. 4, no. 1, 1957.

[4] MASTERMAN, M., R. NEEDHAM and K. JONES: *The Analogy between Mechanical Translation and Library Retrieval,* Int. Conf. on Sci. Inf., Washington D.C., 1958.

6. Discussion

M. M. Astrahan (USA): A machine which can understand the "meaning" of words is essential to an information searching which involves arbitrary questions. It seems, however, that words are not enough and the meaning of words cannot be developed simply by association with other words. Association with recordings of sensory impressions is also needed. For a machine to understand completely the meaning of words it must be equipped to sense the real world as we do, including both external senses such as vision and also internal ones.
There are of course difficulties to be overcome. A very large store with associated access (based on matching the recorded information with the stimulus) will be needed, but even more difficult will be the determination of the logical rules for generalizations and abstract concepts.

R. M. Needham: The classification of word uses in the thesaurus does not give the complete meaning of the word and will fail to distinguish between words which are used in similar verbal contexts, e.g. the names of different kinds of trees. This point, which is the influence of the real world on the system, may be dealt with by a simple listing of 1 — 1 equivalents.

J. *O'Connor (USA):* What is the evidence that the closeness of f(i) distributions between the given and the constructed lattices is a good measure of unlikely accidental agreement? Supposing that as much code compression is possible as the lattice considerations imply, then what is the algorithm and what is the evidence that it can produce a significant amount of compression in a reasonable time?
When can production be expected of a successful compressed coding ?
A possible method of applying this technique of a growing collection would be as follows. New documents coming in could be coded without compression until a block had been accumulated. Then they could be compressed. A search would change codes before entering the new block.

R. M. Needham: The question of the reasonableness of taking the f(i) distribution is excessively detailed. It i derived from the theoretical methods for finding the degree of a lattice, which appear in a paper at present being submitted to a British journal.
The question of the algorithm is also too complex to deal with here.
The programming of the scheme described has been continuing for a year and should be tested by autumn 1959. We have already considered the means of dealing with new accessions suggested by the last speaker and they will be tried when the whole scheme is tested.