m

The Proper Place of Men and Machines in Language Translation

by Martin Kay

CSL-80-11 **October 1980**

Abstract: The only way in which the power of computers has been brought to bear on the problem of language translation is machine translation, that is, the automation of the entire process. Machine translation is an excellent research vehicle but stands no chance of filling actual needs for translation which are growing at a great rate. In the quarter century during which work on machine translation has been going on, there has been considerable progress in relevant areas of computer science. However, advances in linguistics, important though they may have been, have not touched the core of this problem. The proper thing to do is therefore to adopt the kinds of solution that have proved successful in other domains, namely to develop cooperative man-machine systems. This paper proposes a *translator's amanuensis,* incorporating into a word processor some simple facilities peculiar to translation. Gradual enhancements of such a system could eventually lead to the original goal of machine translation.

A version of this paper will appear in *Statistical Methods in Linguistics.*

CR Categories: 3.42.

Key words and phrases: Natural Language, Machine Translation.

XEROX
PALO ALTO RESEARCH CENTER
3333 Coyote Hill Road / Palo Alto **/ California 94304**

INTRODUCTION

The world is badly in need of translators. Almost nobody denies this. The number of pairs of languages between which translations must be made and the number and types of documents involved is constantly increasing. There is not enough money to invest the profession with the status that would attract more people to it and that it certainly deserves. But we are fortunate to be children of the age of computers and it is to them that we naturally turn. A computer is a device that can be used to magnify human productivity. Properly used, it does not dehumanize by imposing its own Orwellian stamp on the products of the human spirit and the dignity of human labor but, by taking over what is mechanical and routine, it frees human beings for what is essentially human. Translation is a fine and exacting art, but there is much about it that is mechanical and routine and, if this were given over to a machine, the productivity of the translator would not only be magnified but his work would become more rewarding, more exciting, more human. It is altogether right that we should look to the computer. Indeed, if the need for translation is as great as it is said to be, the computer is our only hope.

When the computer is improperly used, its effects are, of course, quite different. This happens when the attempt is made to mechanize the non-mechanical or something whose mechanistic substructure science has not yet been revealed. In other words, it happens when we attempt to use computers to do something we do not really understand. History provides no better example of the improper use of computers than machine translation. Of the impressive list of exploits that computer scientists and computational linguists have engaged in over the past twenty years, the only one that has ever succeeded in firing the imagination of translators and their employers has been machine translation. But here the success of machine translation ends. It fires the imagination but it does not, except under very special circumstances, produce useful results.

The late Bar Hillel, a most vociferous critic of machine translation, characterized the ideal towards which inventors in this field strive as *Fully Automatic High-Quality Translation (FAHQT).* The machine would be one which, without human intervention except perhaps at the input keyboard, could render a more or less arbitrary text in one language into a text of equal quality in another. It is surely a worthy ideal and one which has attracted a regrettably small number of linguists and computer scientists. Even if it is never achieved, it provides an incomparable matrix in which to study the workings of human language. Whether it is achieved or not, other useful, if more modest, inventions may well emerge as by-products of the attack on FAHQT provided only that the work is conducted in a healthy intellectual environment. The trouble is that no such environment exists today.

To understand language is to understand how it *works.* Enlightening remarks about it will therefore often be best expressed in terms of *processes.* I take this to mean that important contributions to linguistics are likely to be more and more in the spirit of Artificial Intelligence. A manifesto for this point of view is out of place here. What is to the point is that translation embraces every facet of language while providing a task whose criteria of success, for all their problems, are remarkably well defined. In science in general, and artificial intelligence in particular, the proper rôle of computers is quite different from any they can play in engineering or any enterprise directed towards the fulfillment of immediate and practical needs. Here they are properly applied to what is not understood with the expectancy that, as much by their frequent and resounding failures as by anything else, they will illuminate the boundaries of our ignorance. Engineering pays heavily for the very failures that science can best profit from.

The need for translated texts will not be filled by a program of research that devotes all of its resources to a distant ideal, and linguists and computer experts will be denied the proper rewards of their labors if they must promise to reach the ideal by some specific time. A healthy climate for FAHQT will be one in which a variety of different though related goals are being pursued with equal vigor for the intellectual and practical benefits that they may bring.

There was a long period—for all I know, it is not yet over—in which the following comedy was acted out nightly in the bowels of an American government office with the aim of rendering foreign texts into English. Passages of innocent prose on which it was desired to effect this delicate and complex operation were subjected to a process of vivisection at the hands of an uncomprehending electronic monster that transformed them into stammering streams of verbal wreckage. These were then placed into only slightly more gentle hands for repair. But the damage had been done. Simple tools that would have done so much to make the repair work easier and more effective were not to be had presumably because of the voracious appetite of the monster, which left no resources for anything else. In fact, such remedies as could be brought to the tortured remains of these texts were administered with colored pencils on paper and the final copy was produced by the action of human fingers on the keys of a typewriter. In short, one step was singled out of a fairly long and complex process at which to perpetrate automation. The step chosen was by far the least well understood and quite obviously the least apt for this kind of treatment.

Government and bureaucracy may be imbued with a sad fatalism that forces it to look to the future as destined to repeat the follies of the past, but we can surely take a moment to wonder at the follies of the past and nostalgically to muse about what a kinder and more rational world would be like.

The case against machine translation as a solution to practical problems is overwhelming and has been made many times. I do not propose to repeat it in any detail here. It will, however, be worth a few words to make a *prima facie* case for the implausibility of practical machine translation if only so that the contrast with realistic approaches to the problem will be more striking. I will go on to outline what some of these might be. I shall make some specific proposals, but I should like it to be clearly understood that I do not believe that they represent the only course to follow. They are intended only to illustrate my main point which is this: There is a great deal that computer scientists and linguists could contribute to the practical problem of producing translations, but, in their own interests as well as those of their customers, they should never be asked to provide an engineering solution to a problem that they only dimly understand. By doing only what can be done with absolute surety and reliability now and by going forward from there in short, carefully measured steps, very considerable gains can be virtually guaranteed to all concerned.

The *Prima Facie* Case Against Machine Translation

It is not difficult to convince oneself that fully automatic machine translation is no more a serious answer to any practical problem today than it ever was. But to do so responsibly and scientifically requires examination of a lot of evidence and the careful design and performance of a number of experiments. It is clearly pure irresponsibility to attempt to assess any particular translation system on the basis of intuitive reactions to a so-called *demonstration* in which one examines what the line printer delivers and listens to ingenious attempts to explain the tenuous relationship this bears to, say, English. But it is reasonable and easy to consider what *prima facie* case can be made just on the basis of the advertising. This section contains two related arguments against the plausibility of machine translation as an industrial enterprise, one from the point of view of linguistics and the other from that of computer science.

Machine Translation and Linguistics

Let us look for a moment at a particular problem—one of the prodigious set that the designer of any machine-translation system has to face. Almost any member of the set would serve my purpose, so I will take one of the oldest and most hackneyed, namely how to choose a translation for a pronoun. To state it in this way is, of course, already a gross simplification because no translator or translation system worth its salt choses translations for pronouns or, for that matter, any other isolated words. But the simplification will take nothing from the very elementary point I want to make.

Consider the following pair of sentences:

> *Since the dictionary is constructed on the basis of the text that is being processed, it need refer to only a small amount of context to resolve ambiguities.*

> *Since the dictionary is constructed by a native speaker of the language,*
> *he need refer to only* a *small amount of context to resolve ambiguities.*

Suppose that these are both translations from some other Indo-European language so that, in all probability, the underlined *it* and *he* correspond to the same word in the original in French, the word would be *il*, for example. Now, it is entirely possible that automatic translation systems could be found that would get the translations of both sentences right. But it is almost inconceivable that one could be found that would get them right for the right reasons or that it would systematically solve problems of this kind correctly. If such a system did exist, we could not expect to find its designer in a gathering of this kind because he would surely be *a* person of such saintly modesty and so retiring a nature as to prevent his ever making his results known to others. He would die in poverty and obscurity. A person with normal human weaknesses who had the key to this problem could confidently be expected to claim the crown that linguistics is eager to bestow on him. Pronominal reference is, after all, among the most vexing problems in linguistics. Much the same can be said of innumerable other problems on whose solution the success of machine translation turns. If any of them had in fact been solved, we should not have to purchase an expensive system to find out about it and commercial or proprietary interest would not long hide it from us.

We are forced to one of two conclusions. Either some essentially ad *hoc* solution to these difficulties has been found and built into the systems that are offered for sale, or these systems do not really solve the problems at all. *Ad hoc* solutions tend to be based on case-by-case analyses of what linguists call *surface* phenomena, essentially strings of words, and on real or imagined statistical properties of particular styles of writing and domains of discourse. In scientific and technical texts, for example, one runs less risk of error by translating the French pronouns i/ and *elle* as *it,* rather than *he* and *she,* especially in contexts like *il est possible que....* In // *est convaincu que...,* on the other hand, *he* is a better bet. These facts are listed in the functional equivalent of a dictionary of *words in context* to which new entries are continually brought. The cash value of each new addition is slightly less than that of the one before as the contribution to the device as a whole slowly approaches its asymptote.

In fact, such little documentary evidence as the proprietors of past machine-translation systems have been prepared to release has typically pointed with evident pride to the great number of *ad hoc* devices that they contain and has made the incontestable point that any enhancement of the system in the future will require more and more and more of the same. I will come shortly to the question of what is wrong with engineering products that rest on *ad hoc* devices rather than sound theory.

Machine Translation and Computer Science

The *prima facie* case against operational machine translation from the linguistic point of view will be to the effect that there is unlikely to be adequate engineering where we know there is no adequate science. A parallel case can be made from the point of view of computer science, especially that part of it called *Artificial Intelligence.* To translate is to re-express in a second language what has been understood by reading a text. Any purported solution to the problem that does not involve understanding in some sense is, at best, ad *hoc* and therefore subject to the linguistic objections already alluded to. A large part of the field of artificial intelligence is given over to building models on the basis of which to attempt some explication of this notion of understanding but no serious worker in this field has ever claimed to be able to provide the theoretical support required by any practical enterprise, least of all one so embracing as language translation.

There are also some points about past performance that deserve to be made from the computer scientist's point of view. There is, for example, the question of programming style and technique. The designers of machine translation systems have been intensely concerned with a property of their programs they call *efficiency.* Here is how the argument goes: These systems would not be required at all if there were not large quantities of text to be translated so that if one program took only slightly more computer time than another, it could soon involve a great deal of extra cost when put into operation. This is the main justification for the fact that there has been little or no use of higher-level programming languages. The lower-level assembly languages that have been used give the programmer direct access to the most basic facilities in his machine so that they cannot be less powerful than higher-level languages. Given a program in a higher-level language, it is almost always possible to produce a translation into assembly language that requires less machine time to run. This is not to say that it will not be a difficult, error-prone, and time-consuming operation to do so. Against this obvious advantage of assembly languages must be set their equally obvious disadvantage, namely that they are arch-enemies of clarity and perspicuity. My claim will be that a program written in assembly language is much more likely to embody an *ad hoc* solution to a problem than one written in a higher-level language. This is only to be expected. Assembly languages give equal status to every detail in the specification of the program so that there is no way in which the overall plan that the program embodies can emerge. Consequently, a program that would seem simple in another language is almost guaranteed to look bewilderingly complex in assembly language. A program such as machine translation would require, one that would be complex by any imaginable standard, would be beyond imagination in assembly language. In programming, as in any kind of writing, the most complex ideas require the greatest clarity and skill for their exposition. But programming differs from everyday communication in that the languages available differ greatly in expressive power and the choice among them severely conditions the clarity that can be achieved. Every computer scientist is taught, but only comes truly to appreciate as a result of bitter experience, that programs are written for a human as well as a mechanical audience and the most important member of that audience is himself. A programmer who writes in assembly language is necessarily giving us less than his best at the highest possible price.

Efficiency, in computer programming, is itself a complex and subtle matter. It is true that it is affected by such issues as the language that a program is written in, but these effects are, at worst, linear. More realistically speaking, they are sublinear because a very large proportion of the time taken for any large program to run is accounted for by a very small proportion of the code. Standard practice, therefore, is to write a program in a language that displays its structure as clearly as possible and to rewrite carefully selected small portions of it in assembly language only when experience has clearly demonstrated that the effort involved in doing this would amply repay the effort.

Truly significant gains in efficiency invariably come from adjustments to the algorithm itself, that is, to the overall strategy that the program employs. Consider a simple example. The words of a text are to be looked up in a dictionary. There are a great many strategies that could be used, all of which would produce identical results for the same words and the same dictionary, but at very different cost in machine time. The dictionary could be searched from the top for each separate word in the text. Binary search, hashing, or one of the innumerable variants of these could be used. A method that has been popular with the designers of machine-translation systems is to sort the words of the text into alphabetical order so that a single pass through the resulting list and the dictionary locates all relevant entries. These are then sorted back into the order of the text.

Quite independently of the machine or the programming system used to implement them, these techniques can all be analyzed in terms of the way the time they take is related to the length of the text and the size of the dictionary. If there are m entries in the dictionary and n words in the text, if the dictionary is not ordered in any especially helpful way, and if almost all the words are in the dictionary, then the first method requires each of the n words in the text to be compared with about half the words in the dictionary so that the time involved will vary with both m and n, in other words, it will be proportional to mn. Putting the dictionary in order by frequency could conceivably improve things to the point where the average word is found in the first log m entries, which would make the technique as efficient as binary search. A suitably chosen hashing scheme removes the effect of m altogether so that, at least from this point of view, this method is better than either of the others. The technique that requires sorting proceeds in three steps, two sorts and the comparison with the dictionary. The comparison with the dictionary is linear, but sorting n items takes on the order of $n \log n$ steps by the best known methods, which means that the time taken by the comparison is not significant. This is a simple classic case where the considerations determining the best solution are well known. In reality, the choice is often difficult and text-book solutions are not available.

There is a branch of computer science called *analysis of algorithms* that is devoted to the assertions of this kind that can be made about computational methods. What is important about such assertions is that they characterize the cost of a technique *as a function of* the data it will be applied to. Differences in the functions that characterize competing techniques are altogether more significant than the purely linear differences that programming languages and coding practice can affect. If techniques *A, B*, and *C* all achieve the same results when applied to an input of size s, and the time taken by *A* varies with s^2, *B* with s, and *C* with $\log s$, then *C* is best and *A* is worst and, unless s is very small, the implementation details are beside the point. If *C* takes 10 steps for a certain case, then *B* will take about 1000, and *A*, 1,000,000. When the differences are as great as these—and they often are—the cost of the individual steps is irrelevant.

Any program that purports to translate natural text must clearly be orders of magnitude more complex than one that simply looks words up in a dictionary. It will always be susceptible of improvement, at least in a theoretical sense, not only in the quality of the results it delivers but also in the efficiency of the algorithms it incorporates. To be continually improvable in this way, a program must be perspicuous and robust. It must be perspicuous so that there is never any doubt about the role that each of its parts plays in the overall structure and robust so that it can be changed in important ways without fear of damage. Perspicuity and robustness are clearly two sides of the same coin. They are the high ideals to which the art of programming is continually striving and which it never achieves.

The Statistical Defense

It is immediately clear why *ad hoc* solutions should be offensive to a scientist. His job is, in a sense, precisely to reveal as principled and orderly what had previously been ad hoc. But what we must attend to is whether these solutions should upset an engineer or someone whose primary concern is getting a job done and, if so, to what extent. Two arguments are commonly made for *ad hoc* solutions to the problems of machine translation. The first is a simple statistical claim that can be dismissed almost as easily as it can be stated. The second is what I shall refer to as the *sorcerer's apprentice* argument.

The statistical argument rests on the fact that something can be complex and subtle without the complexities and subtleties being spread uniformly through it. Linguistics requires of its practitioners remarkable virtuosity in constructing examples of problems such as no existing or proposed computer system could possibly solve. But the claim is that we do not have to solve them so long as they do not crop up very often. We may not have an algorithm that will identify the antecedent of a pronoun whenever a human reader could but, if we can devise a method that will identify it most of the time, that will be good enough.

An algorithm that works most of the time is, in fact, of very little use unless there is some automatic way of deciding when it is and when it is not working. If it were able to draw a proofreader's attention to all the cases of pronominal reference that were in doubt so that these, and only these, would have to be examined by a human reader, and if a high proportion of the cases were known to be correctly handled, then the utility of the technique would be clear. But the statistical argument is usually stated in the weaker form.

Suppose that a good, reliable translation of a text is required and that a computer program is available that translates pronouns correctly 90 per cent of the time. If there were some way to tell which 10 per cent of the pronouns had been wrongly translated, it would be sufficient to examine these to verify the correctness of the translation (ignoring, for simplicity, other possible sources of error). But since this cannot be done, 100 per cent of the pronouns must be examined. To find a pronoun and check that it is correctly translated is expensive relative to making the correction. Therefore, it does not matter very much if the program is right 90, 99, 80, or 50 per cent of the time. The amount of work that it leaves for the repairman is essentially the same. Somebody may claim, however implausibly, that 10 per cent of pronouns occur in contexts where the translation is not crucial. This would be a useful thing to know just in case these were precisely the instances that the machine translated incorrectly but no such argument has been, or is likely to be, upheld.

The real situation is much worse because there is more to translation than pronouns. A great many decisions of essentially the same difficulty must be made in the course of translating a single sentence. If there is reason to expect each of them to be correct 90 per cent of the time, there need only be seven of them in a stretch of text to reduce the expectation of translating it correctly to below 50 per cent.

The moral is clear. The overall efficiency of a translation system, human or electronic, is directly related to its reliability. If it falls short of the acceptable standard, *to any degree whatsoever,* it might as well fail grossly because the burden it places on the proofreader will be very large, and not notably different in either case. The efficiency of a translation system, like any other, must be assessed over all its components, human and mechanical.

The Sorcerer's Apprentice Defense

The Sorcerer's Apprentice argument is to the effect that the kind of incomplete theory that linguists and computer scientists have been able to provide is often a worse base on which to build practical devices than no theory at all because the theory does not know when to stop. When a theory proposes questions about the data to which it can provide only partial answers, it is often better that the question should never have been asked.

Consider the following version of an often quoted sentence:

The man looked at the girl with the telescope

It will be pointed out that this can be translated word-for-word into French, and innumerable other languages, and gives a perfectly adequate result. It is, of course, ambiguous in various ways because of the different roles that the prepositional phrase can play in the syntactic structure. But French admits exactly parallel ambiguities so that any effort spent trying to decide whether the girl had the telescope or the man had it and used it to see her with, is wasted. In fact, such an effort can serve only to jeopardize the translation because, if it results in any but a word-for-word translation of this sentence, there is an unnecessary risk that it will be wrong.

On the other hand, if the sentence had been

The man looked at the girl with penetrating eyes

the question of whose eyes were involved would suddenly have been important because no acceptable word-for-word translation is possible; we are forced to choose between *aux yeux* and *de ses yeux. Avec* is a good translation for *with* in neither case. What algorithm will tell a translator that this case needs analysis whereas the first one does not? Perhaps the absence of an article before *penetrating eyes* gives the clue. This would indicate that

He looked at the girl with affection

requires analysis. Unfortunately for the argument, it does not.

The main problem with the sorcerer's-apprentice argument is that the decision that a sentence could be translated without analysis can only be made after the fact. Analysis shows that there is more than one interpretation of a sentence at some level and further analysis shows that there is a single translation that is compatible with each of them. In short, the algorithm required to decide when analysis is required would have to use the results of the very analysis it is designed to avoid.

What the sorcerer's-apprentice argument does suggest is that the process of translation should proceed in the following nondeterministic fashion. Whenever the information needed to make a choice reliably is not available, all possibilities should be followed up independently. Furthermore, when an essentially arbitrary choice must be made, say between a pair of synonymous words, these possibilities should also be held open. Under this policy, a given sentence of input would yield a family of sets of sentences in the target language. The members of each set are presumed equivalent and the sets are distinguished by the different patterns of decisions that led to their production. If, by happy chance, there is a sentence that belongs to every set in the family, then it is presumably the safest, and possibly even the best, translation.

Consider, for example, the following somewhat contrived French sentence:

Ils signeront le document pourvu que leur gouvernement accepte.

Possible translations, classified by family, are

I

They will sign the document supplied that their government accepts.

They will sign the document furnished that their government accepts.

They will sign the document provided that their government accepts.

They are going to sign the document supplied that their government accepts.

They are going to sign the document furnished that their government accepts.

They are going to sign the document provided that their government accepts.

etc.

II

They will sign the document provided that their government accepts.

They will sign the document on condition that their government accepts.

They will sign the document only if their government accepts.

They are going to sign the document provided that their government accepts.

They are going to sign the document on condition that their government accepts.

They are going to sign the document only if their government accepts.

etc.

The two translations come from two quite different analyses of the original. It could be only as a result of a quite remarkable chance that a pair of interpretations as different as these should fall together. They would not have done so, for example, if *accepter* had been a verb that showed a difference between its indicative and subjunctive forms or if a feminine or a plural noun had taken the place of *document.* However, since the sentences involving the phrase *provided that* belong to both sets, the choice of a translation can be narrowed to them because this neutralizes the ambiguity.

This technique does not depend on there being a sentence that appears in every member of the family. Whenever a single sentence occurs in more than one set, they can be reduced to a single set containing only the intersection of the originals. There are optimal ways of choosing sets to conflate so as to reduce the choice that must eventually be made to a minimum. Furthermore, it is not difficult to devise extensions of the procedure. If the sets in the family of translations are labeled in some way for the places in the analysis where a decision was made in the course of their production, then the differences between pairs of translations can be ascribed to specific sets of decisions. If there is no translation that belongs to all the sets, then the number and the difficulty of the decisions that need to be made to make the choice can be minimized. This is a topic I shall return to. For the moment, the point to note is that the observation on which the sorcerer's apprentice argument is based tends to maximize the amount of computation to be done • just the inverse of their original intent.

THE TRANSLATOR'S AMANUENSIS

I come now to my proposal. I want to advocate an incremental approach to the problem of how machines should be used in language translation. The word *approach* can be taken in its original meaning as well as the one that has become so popular in modern technical jargon. I want to advocate a view of the problem in which machines are gradually, almost imperceptibly, allowed to take over certain functions in the overall translation process. First they will take over functions not essentially related to translation. Then, little by little, they will approach translation itself. The keynote will be modesty. At each stage, we will do only what we know we can do reliably. Little steps for little feet!

Text Editing

The easy way to prepare a piece of text is the way this one was prepared, that is, with a text-editing program on a computer. It does not matter whether it is done on a very small and personal computer that fits under the table in your office or on a large time-sharing machine, except that the latter is apt to be expensive. It matters very much that the design of the editor should be in the best possible taste, and it makes some difference whether the facilities include a screen that the writer can point at when he wishes to draw the program's attention to a particular place. People who have worked with bad editors soon retreat to the security of their typewriter or a pencil; anyone who has worked with a good one cannot be dragged away with a team of wild horses.

So, one thing to do would be to get a good editor and give it to your translators. If you could only do one thing, this would probably be the best. But you would do better to find an inventive computer scientist with good taste and to have him design a special editor which, in its earliest incarnations, would do little more than the program he would design for anyone else. But the design would be flexible and make provision for various kinds of extension. The kind of computer scientist I have in mind will expect to see the initial product in operation for a while before he makes detailed proposals for the extensions and he will probably want to see various alternative forms of each extension in use before any one is adopted.

Let us be specific. The device I am about to describe, which I call *The Translator's Amanuensis* does not exist and probably never will. It is not the result of a careful program of design so that its details are ill specified, and what is specified I have invented only to illustrate the kind of avenue that seems most fruitful to follow and to avoid a long sequence of conditional sentences I should otherwise have to write.

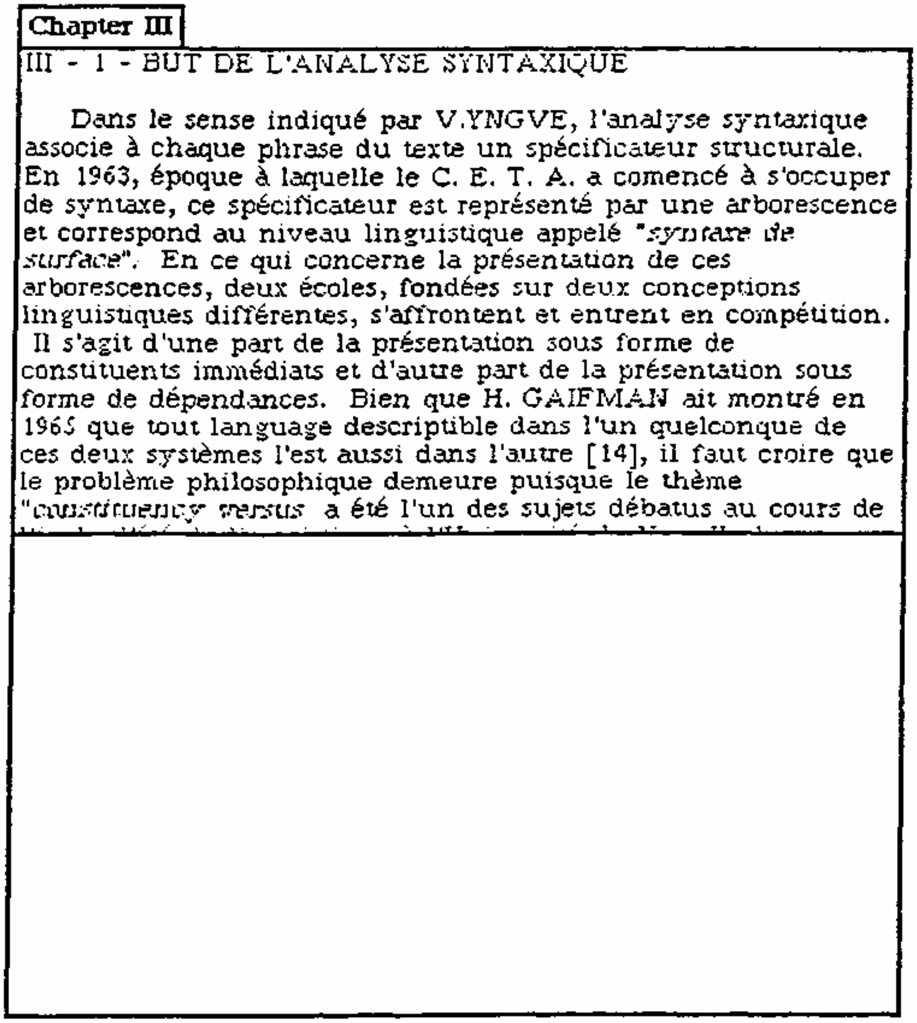
Suppose that the translators are provided with a terminal consisting of a keyboard, a screen, and some way of pointing at individual words and letters. The display on the screen is divided into two windows. The text to be translated appears in the upper window and the translation will be composed in the bottom one. Fig. 1 shows how the screen might appear before the translation process begins. Both windows behave in the same way. Using the pointing device, the translator can *select* a letter, word, sentence, line, or paragraph and, by pressing the appropriate key, cause some operation to be visited upon it.

There are various styles of work that a translator might adopt using this device. One that I shall pursue briefly here involves first copying the entire text to be translated into the bottom window. It thereby becomes, so to speak, the first draft of the translation. Little by little, words, phrases and sentences will be replaced by true translations until, in the end, little or nothing of the original remains in the bottom window. This somewhat unconventional procedure has the advantage of making it possible for the machine to maintain detailed linkages between the original and the translation so that it has a detailed idea of what corresponds to what.

In Fig. 2, the words *indicated by* have been selected because it has been decided that they constituted too literal a translation. The translator now gives the REPLACE command, say by striking R on the keyboard, and the selected word is replaced by a symbol showing that subsequent characters will be accepted as a new insertion at that place. In this case, the translator types *of* and the display is adjusted to show the amended text.

Translation Aids

This recital could be continued indefinitely. Basically, what I am describing is an editor of a kind that has become quite common. Now, let us consider how this device might be made to give special service to a translator. In line with the incremental approach, I will start



Chapter III

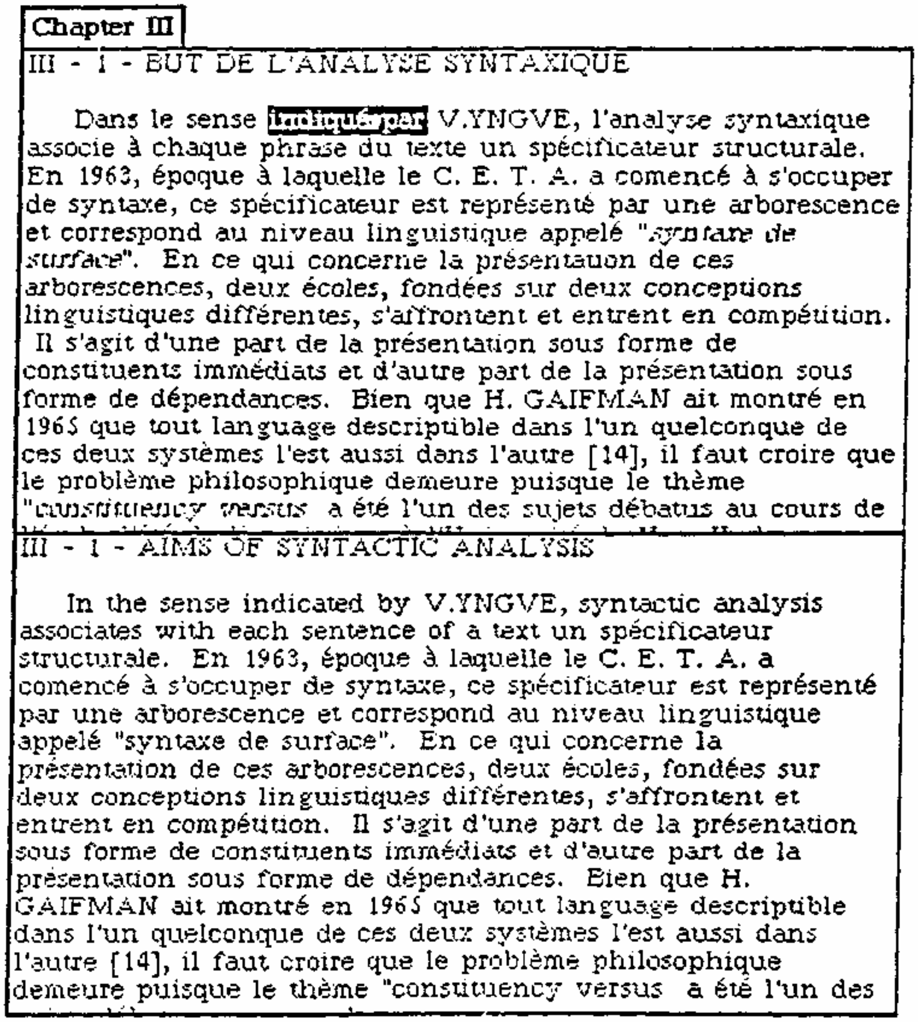
III - 1 - BUT DE L'ANALYSE SYNTAXIQUE

Dans le sense indiqué par V.YNGVE, l'analyse syntaxique associe à chaque phrase du texte un spécificateur structurale. En 1963, époque à laquelle le C. E. T. A. a comencé à s'occuper de syntaxe, ce spécificateur est représenté par une arborescence et correspond au niveau linguistique appelé "*syntaxe de surface*". En ce qui concerne la présentation de ces arborescences, deux écoles, fondées sur deux conceptions linguistiques différentes, s'affrontent et entrent en compétition. Il s'agit d'une part de la présentation sous forme de constituents immédiats et d'autre part de la présentation sous forme de dépendances. Bien que H. GAIFMAN ait montré en 1965 que tout language descriptible dans l'un quelconque de ces deux systèmes l'est aussi dans l'autre [14], il faut croire que le problème philosophique demeure puisque le thème "*constituency versus* a été l'un des sujets débatus au cours de

Fig.1—The Initial Display

simple. A relatively trivial addition would be a dictionary. The translator selects a word or sequence of words and gives a command to cause them to be looked up. In Fig. 3, the word *spécificateur* has been selected. When the lookup command is given, a new smaller window appears at a place indicted by the user. This new window gives the effect of overlaying some portion of the windows already present. In this case, the new window contains a deceptively simple dictionary entry for the selected word.

The simplicity of the dictionary entry is a feature of the system. We should think of the dictionary that the system has on file as being large and highly structured, growing on a daily basis as its deficiencies are revealed. To consult an entry, the user of the system is



Chapter III

III - 1 - BUT DE L'ANALYSE SYNTAXIQUE

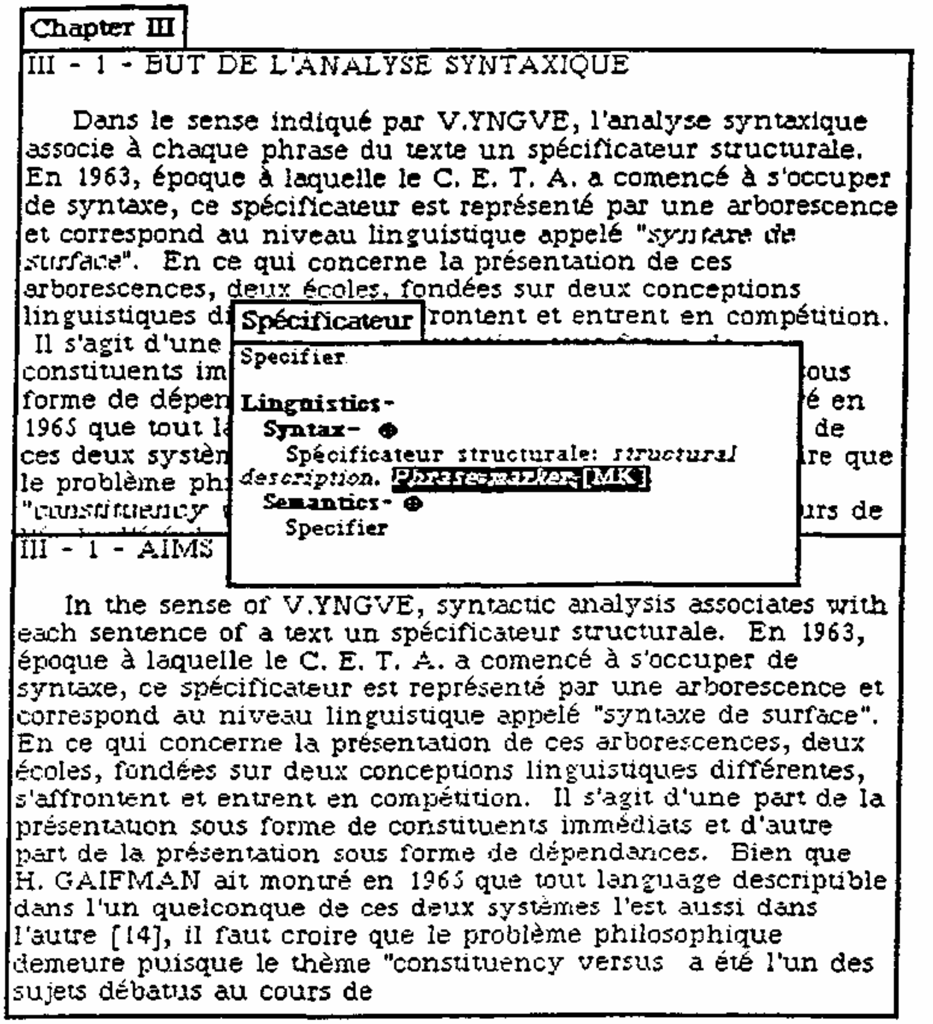
Dans le sense indiqué par V.YNGVE, l'analyse syntaxique associe à chaque phrase du texte un spécificateur structurale. En 1963, époque à laquelle le C. E. T. A. a comencé à s'occuper de syntaxe, ce spécificateur est représenté par une arborescence et correspond au niveau linguistique appelé "*syntaxe de surface*". En ce qui concerne la présentation de ces arborescences, deux écoles, fondées sur deux conceptions linguistiques différentes, s'affrontent et entrent en compétition. Il s'agit d'une part de la présentation sous forme de constituents immédiats et d'autre part de la présentation sous forme de dépendances. Bien que H. GAIFMAN ait montré en 1965 que tout language descriptible dans l'un quelconque de ces deux systèmes l'est aussi dans l'autre [14], il faut croire que le problème philosophique demeure puisque le thème "constituency versus a été l'un des sujets débatus au cours de

III - 1 - AIMS OF SYNTACTIC ANALYSIS

In the sense indicated by V.YNGVE, syntactic analysis associates with each sentence of a text un spécificateur structurale. En 1963, époque à laquelle le C. E. T. A. a comencé à s'occuper de syntaxe, ce spécificateur est représenté par une arborescence et correspond au niveau linguistique appelé "syntaxe de surface". En ce qui concerne la présentation de ces arborescences, deux écoles, fondées sur deux conceptions linguistiques différentes, s'affrontent et entrent en compétition. Il s'agit d'une part de la présentation sous forme de constituents immédiats et d'autre part de la présentation sous forme de dépendances. Bien que H. GAIFMAN ait montré en 1965 que tout language descriptible dans l'un quelconque de ces deux systèmes l'est aussi dans l'autre [14], il faut croire que le problème philosophique demeure puisque le thème "constituency versus a été l'un des

Fig.2—Selection

therefore provided with special tools. He is first shown only a gross summary of what the entry contains. By pointing to a subentry in that summary, he can obtain information on the next level of structure in a new window. The strange symbols following the words *Syntax* and *Semantics* in Fig. 3 represent text which will be included if the user points to them. The text that then appears may contain other instances of this symbol, and so on. The translator can thus cause the entry to develop in the direction indicated by the text on hand. At any time, he can return to a higher level by pointing at some part of the corresponding window that still remains exposed. If, in the course of translating *a* text, a word or phrase is looked up a second time, the display will show, not just the top-level entry, but the situation that



Chapter III

III - 1 - BUT DE L'ANALYSE SYNTAXIQUE

Dans le sense indiqué par V.YNGVE, l'analyse syntaxique associe à chaque phrase du texte un spécificateur structurale. En 1963, époque à laquelle le C. E. T. A. a comencé à s'occuper de syntaxe, ce spécificateur est représenté par une arborescence et correspond au niveau linguistique appelé "*syntaxe de surface*". En ce qui concerne la présentation de ces arborescences, deux écoles, fondées sur deux conceptions
linguistiques di ... rontent et entrent en compétition.
Il s'agit d'une
constituents im ... ous
forme de dépen ... é en
1965 que tout l ... de
ces deux systèn ... re que
le problème ph
"constituency ... urs de

Spécificateur

Specifier

Linguistics-
Syntax- ⊕
Spécificateur structurale: *structural description.* *Phrase-marker* [MK]
Semantics- ⊕
Specifier

III - 1 - AIMS

In the sense of V.YNGVE, syntactic analysis associates with each sentence of a text un spécificateur structurale. En 1963, époque à laquelle le C. E. T. A. a comencé à s'occuper de syntaxe, ce spécificateur est représenté par une arborescence et correspond au niveau linguistique appelé "syntaxe de surface". En ce qui concerne la présentation de ces arborescences, deux écoles, fondées sur deux conceptions linguistiques différentes, s'affrontent et entrent en compétition. Il s'agit d'une part de la présentation sous forme de constituents immédiats et d'autre part de la présentation sous forme de dépendances. Bien que H. GAIFMAN ait montré en 1965 que tout language descriptible dans l'un quelconque de ces deux systèmes l'est aussi dans l'autre [14], il faut croire que le problème philosophique demeure puisque le thème "consituency versus a été l'un des sujets débatus au cours de

Fig.3—Looking up Terms

was obtained when the same entry was consulted previously. This is on the theory that the same part of the dictionary is apt to be most relevant. The example given in the illustration is unrealistically simple; we must envisage many levels of structure and a greater investment of effort in the corresponding system design.

The translator can edit dictionary entries with the same commands that he uses for the translation itself. These amendments may be temporary, serving essentially as notes on the vocabulary of the particular document and the terminological decisions that have been made. They can also be more permanent, providing instant information to other translators with similar problems. Communication of this sort, across time as well as space is one of the most crucial functions that computers can serve.

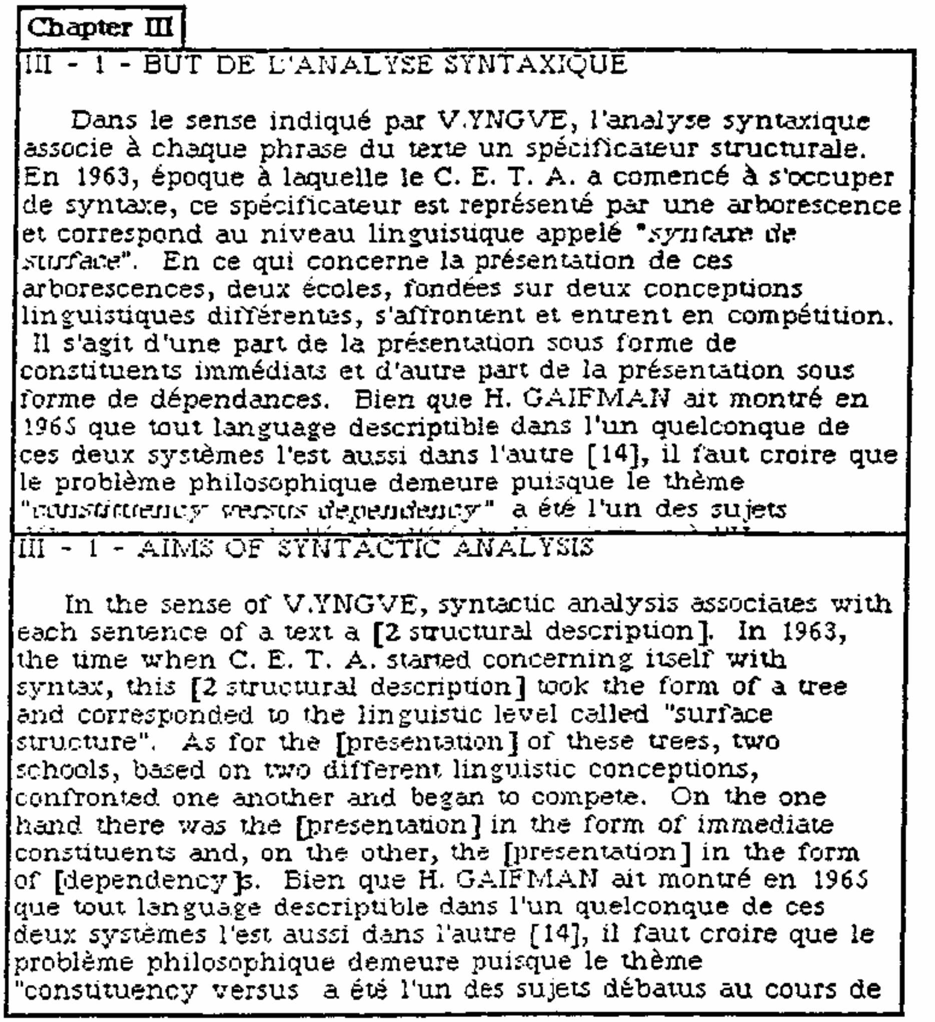
I take it that words selected for reference to the dictionary will not have to be in their citation form. The computer will be able to apply rules of morphological analysis to determine the proper dictionary heading for itself. While this is not trivial, it is one of the few parts of linguistic analysis that is well understood. Furthermore, it should be possible to look up compounded words and sequences suspected of constituting an idiom or fixed phrase.

The machine's dictionary can be used in a variety of ways. Suppose, for example, that a word is put in the local store—that part of the dictionary that persists only as long as this document is being worked on—if it occurs in the text significantly more frequently than statistics stored in the main dictionary indicate. A phrase will be noted if it occurs two or three times but is not recognized as an idiom or set phrase by the dictionary. By examining the contents of this store before embarking on the translation, a user may hope to get a preview of the difficulties ahead and to make some decisions in advance about how to treat them. These decisions, of course, will be recorded in the store itself. In the course of doing this or, indeed, for any reason whatever, the translator can call for a display of all the units in the text that contain a certain word, phrase, string of characters, or whatever. After all, the most important reference to have when translating a text is the text itself.

If the piece of text to be translated next is anything but entirely straightforward, the translator might start by issuing a command causing the system to display anything in the store that might be relevant to it. This will bring to his attention decisions he made before the actual translation started, statistically significant words and phrases, and a record of anything that had attracted attention when it occurred before. Before going on, he can examine past and future fragments of text that contain similar material.

Most editing programs allow the writer to insert an arbitrary symbol of his own choosing at various places in the text and, at some later time, to cause all instances of that symbol to be replaced by some other word or symbol. This comes close to filling an important need that translators have. It turns out that a particularly vexing problem of technical translation is that of vocabulary control. That you should translate a technical term in one language by the proper technical term in the other language is important, but it is less important than that you should translate it always in the same way. One way to achieve this would be to make up a symbol, containing some otherwise unused character, and then to make replacements when the translation was complete. The device envisaged here goes further.

I suppose that the user of the system has available a special pair of brackets that he can insert in the text; in the examples they appear square and bold. They appear on his screen but will not be printed in the final translation. They are used as follows. If it is, for the moment, unclear how a word or technical phrase should be treated, the tentative translation is enclosed in these special brackets. They can be use in the translation itself or



Chapter III

III - 1 - BUT DE L'ANALYSE SYNTAXIQUE

Dans le sense indiqué par V.YNGVE, l'analyse syntaxique associe à chaque phrase du texte un spécificateur structurale. En 1963, époque à laquelle le C. E. T. A. a comencé à s'occuper de syntaxe, ce spécificateur est représenté par une arborescence et correspond au niveau linguisuque appelé "*syntaxe de surface*". En ce qui concerne la présentation de ces arborescences, deux écoles, fondées sur deux conceptions linguistiques différentes, s'affrontent et entrent en compétition. Il s'agit d'une part de la présentation sous forme de constituents immédiats et d'autre part de la présentation sous forme de dépendances. Bien que H. GAIFMAN ait montré en 1965 que tout language descriptible dans l'un quelconque de ces deux systèmes l'est aussi dans l'autre [14], il faut croire que le problème philosophique demeure puisque le thème "*constituency versus dependency*" a été l'un des sujets

III - 1 - AIMS OF SYNTACTIC ANALYSIS

In the sense of V.YNGVE, syntactic analysis associates with each sentence of a text a [2 structural description]. In 1963, the time when C. E. T. A. started concerning itself with syntax, this [2 structural description] took the form of a tree and corresponded to the linguistic level called "surface structure". As for the [presentation] of these trees, two schools, based on two different linguistic conceptions, confronted one another and began to compete. On the one hand there was the [presentation] in the form of *immediate* constituents and, on the other, the [presentation] in the form of [dependency]s. Bien que H. GAIFMAN ait montré en 1965 que tout language descriptible dans l'un quelconque de ces deux systèmes l'est aussi dans l'autre [14], il faut croire que le problème philosophique demeure puisque le thème "consutuency versus a été l'un des sujets débatus au cours de

Fig.4—Morphology and Lexical Alternatives

in dictionary entries. When the same word or phrase turns up again, the bracketed phrase is explicitly copied into the new position, thus maintaining an association among ail the places where it is used. If the contents of such a pair of brackets is changed, the contents of ail the others that are linked to it change automatically in the same way. Notice that this is a considerably finer instrument than the replacement facility of standard text-editing programs because the changes effect only those occurrences of a word or phrase that have been explicitly associated. Furthermore, if inflectional material belonging to one of these bracketed words or phrases is written in a standard, regular form outside the brackets—as in the case of the word *dependency* in Fig. 4—the appropriate form of the word can be constructed when the final version is settled on. Once again, this calls for the application of morphological rules, this time in the generative direction.

Fig. 4 also illustrates another possible variant of this device. If more than one translation is being considered for a particular term, possibly because both are suggested in the dictionary, the fact is recorded by displaying a bold numeral just after the open bracket. If the translator points at this, the next possibility is taken, both in this and the other places in the text that are linked to it. In this way he can rapidly switch back and forth between variants without having to type.

Machine Translation

There is no early limit to the facilities that could, and probably should, be added to the translator's amanuensis. Rather than prolonging the rehearsal, let us look at where the process might end. I began by proposing an incremental approach to machine translation, so it is machine translation that must come at the bottom of the list. But, if it is to avoid the objections made by myself and others, it must be machine translation in a new form.

I propose that one of the options that should be offered to a user of the hypothetical system I have been describing, at a fairly early stage, be a command that will direct the program to translate the currently selected unit. What will happen when this command is given will be different at different stages of the system's development. But a user of the system will always be empowered to intervene in the translation process to the extent that he himself specifies. If he elects not to intervene at all, a piece of text purporting to translate the current unit will be displayed in the lower window of his screen. He will be able to edit this in any way he likes, just as post-editors have done in the past. Alternatively, he may ask to be consulted whenever the program is confronted with a decision of a specified type, when certain kinds of ambiguities are detected, or whatever. On these occasions, the system will put a question to the human translator. He may, for example, ask to be consulted on questions of pronominal reference.

The only difference between the translation facilities of the translator's amanuensis and previous machine-translation systems that can be seen from a user's point of view is that here the translator has his say while the translation is under way whereas previously he had to wait. If this scheme can be made to work, as I claim it can, its many advantages are collectively overpowering.

The kind of translation device I am proposing will always be under the tight control of a human translator. It is there to help increase his productivity and not to supplant him. It will never resort to *ad hoc* measures that have not been explicitly sanctioned by him. In its normal and recommended mode of use, it will appeal to him rather than being forced back on unfounded guesses. After the system has been under development for a while, Its users will either still be using it as a clerical aid, or they will be consigning considerable amounts of the actual translation work to it. The usage will remain mainly clerical only if the best efforts invested in the translation facilities failed to make them useful or economical. In other words, this system will certainly be able to undertake whatever present systems are able to undertake reliably and if that proves to be very little, the inference is clear. But there is reason to hope that it will undertake more. A system that never reached the stage of proposing translations would still be of inestimable value in automatically producing the final copy, looking up words and phrases faster and in a larger and constantly growing dictionary than would be possible any other way, in keeping notes about vocabulary usage and the like.

There are several important reasons to expect better performance of a system that allows human intervention as opposed to one that will brook no interference until all the damage has been done. First, the system is in a position to draw its human collaborator's attention to the matters most likely to need it. It is clearly important that he should give special attention to matters for which the designers of the system were unable to provide satisfactory algorithmic solutions. A wrong answer in these cases does nothing but mislead. It is far better that the labor and ingenuity spent on developing the machine's ability to make bad guesses should be employed more productively.

The second point is related. The decisions that have to be made in the course of translating a passage are rarely independent. The outcome of one decision typically determines whether certain other decisions will have to be faced at all. A wrong decision at the beginning of such *a* chain leads the system to ask questions of the data before it that do not even make sense; whatever answer is given, it is bound to be wrong. Cascading errors of this kind are common in language processing. In the kind of system proposed here, they will not happen except when the human member of the team makes an error, or when he consigns too much to the machine. In the standard case, he will be consulted when the first decision in the chain is reached and will direct the machine along the right lines.

A third point concerns the machine's use of history. One of the most important facilities in the system is the one that keeps track of words and phrases that are used in some special way in the current text. It is a device that should probably be extended in a variety of ways to cover more than just vocabulary usage. By means of this, the translator is able to make a decision on the first occasion that a difficulty arises that will determine how both he and the machine treat it on all subsequent occasions. In other words, the man and the machine are collaborating to produce not only a translation of a text but also a device whose contribution to that translation is being constantly enhanced. A post-editor who changes something at the beginning of a translation must expect to make essentially the same change many more times before he finishes. This is not to say that a machine-translation program could not be devised that modified its behaviour in the light of experience. However, such a system would be especially liable to the last objection made, namely that bad decisions made early lead to worse decisions later. It is bad enough that ill-founded decisions made early in the processing of a sentence should be allowed to engender other ones later; to extend this policy over an entire text is to invite disaster. The system proposed here will accumulate only experience of what was agreed upon between both human and mechanical members of the team, the mechanical always deferring to the human.

The translator's amanuensis will not run before it can walk. It will be called on only for that for which its masters have learned to trust it. It will not require constant infusions of new *ad hoc* devices that only expensive vendors can supply. It is a framework that will gracefully accommodate the future contributions that linguistics and computer science are able to make. One day it will be built because its very modesty assures its success. It is to be hoped that it will be built with taste by people who understand languages and computers well enough to know how little it is that they know.