A case system processor for the PRECIS indexing language

Peter Hancox and Frederick Smith

The PRECIS indexing system has been successfully applied to languages other than English, and the British Library PRECIS Translingual Project devised an interlingual system (not implemented) to translate between English, French and German. An analysis of the system was used in the design of a transfer indirect system presented here to translate from English to French. To resolve the lack of one-to-one correspondences between prepositions, a semantic analysis of the case system or grammar type was imposed on a syntactic analysis of the string. Strings drawn from BTAIS were used to test the performance of the system. The results are presented and the weaknesses analysed.

Introduction
The module herein described is part of a system that translated PRECIS strings from English into French. It is therefore an amalgam of machine translation and information retrieval. Machine translation has been the subject of serious academic investigation for nearly forty years. Its history can be best understood by tracing its treatment of ambiguity. At first the problem of machine translation was seen as a lexical one, with syntax being of little importance. Lexical ambiguity was larger than initially realised with, for instance, the correct translation of a word being dependent on its syntactic category. Further work indicated that the overall syntactic structure of a sentence is significant. Now it is generally realised that a machine translation system capable of producing adequate results must have a semantic analysis component that is capable of resolving ambiguities still remaining after syntactic analysis.

Three strategies for machine translation have evolved over the last forty years (Figure 1). The direct strategy was the earliest. Here a pair of languages is translated, with no more processing being done than is needed to translate from the particular source language into the particular target language. Obviously for each language pair (in each direction), a new system has to be produced. The indirect approach is divided into two. Interlingual indirect systems translate from the source language into the interlingua (abstract language) and from that into the target language or languages. The transfer indirect system produces an analysis of the source language to an abstract level. The output of analysis is fed into one or more transfer modules, which are specific to each language pair (in each direction). These modules produce rough and ready translations in the target language, which is refined and polished by the generation (also known as the synthesis) module (19).

As will be seen from Figure 1, the direct strategy is by far the least economic when several languages are involved. Superficially the
Figure 1
Strategies for machine translation
interlingual approach seems the most efficient, but the Figure does not illustrate the difficulty of constructing an adequate interlingua capable of expressing the full range of several natural languages. Therefore the transfer indirect strategy represents the most economic approach where several languages are likely to be included. Obviously a system that translates only in one direction, e.g. English to French, can have the transfer architecture, without other languages having to be added.

On the information retrieval side, PRECIS grew out of research carried out in the sixties for a new general facetted classification. Though not a classification, it uses a classificatory feature in the form of role operators to organise terms drawn from natural language into "context-dependent" order. It was designed specifically to provide a printed index to the classified sequence of the British National Bibliography. In manipulating a string of terms to provide a set of coextensive subject statements under several lead-ins, the machine program preserves that original context-dependent order by a process called shunting and, thereby, meanings remain free from ambiguity in the greater percentage of all subject entries. To resolve problems of ambiguity in the remainder and because PRECIS also recognises grammatical structures which may loosely be called predicates, it incorporates a facility which enables the indexer to preserve a closer linkage between some pairs, or larger groups, of terms by connectives usually in the form of propositions. There is also a facility to add substitute phrases to re-express terms in a more natural way when they are to provide part of the context to an entry.

Natural curiosity led those responsible for the development of PRECIS within the British National Bibliography to attempt its application to languages other than English. The conclusion of this was the publication in the PRECIS Manual of an exemplary string in ten languages (1). This exercise demonstrated that the order of terms in a string (as organised by the role operators) need not change between languages. Obviously the vocabulary is individual to each language, as it is drawn from associated natural language.

Others were encouraged by this work, and in the remainder of the decade, there were several experimental studies of the application of PRECIS in French (13,21,22,23), in German and the Germanic languages (30,31), and in Swedish and Danish. More recently, the National Film Board of Canada has begun to issue bilingual versions of its catalogue, with PRECIS indexes separately created in English and French. Austin has demonstrated in his PhD thesis (2) that PRECIS is a multilingual system, being neutral as to the language from which it draws its vocabulary.

The British Library PRECIS Translingual Project
PRECIS was designed to be integrated with the MARC records, which in turn came to be seen as a vessel for the international communication of bibliographic information. The language barrier raised its ugly head at this point because, as regards the British Library data, the subject information was tied to English. The availability of MARC records on international networks such as the B/R's Ruronet just served to
emphasise the problem. So translation became to be seen as not desirable, but necessary.

This then was the environment of the Project. Its orientation owed much to the ideas expressed in the Manual (1). The Reference Indicator Numbers suggested a number language or interlingua and this, coupled with the techniques for organising multilingual thesauri led the proposers of the Project to concentrate on an interlingual approach.

This Project was funded by the British Library Research and Development Department, to run from 1976 to 1979. In the public announcement of the funding, the general aims were stated as:

'...to create a set of routines and computer programs which will add a translational component to the PRECIS system. This will enable the computer to convert the input string into a series of language-independent codes and translate these later into appropriate terms in a target language. These terms will then be manipulated into index entries in the target language without further intervention by the indexer' (9).

Unfortunately, computer support became unavailable and, rather than implementation, the project team had to be satisfied with the design of detailed specifications for all translational procedures (set down in the form of flowcharts), and the manual testing of these algorithms in a manner that would simulate (as far as was reasonable) the decision-making procedures of a machine system (33).

The methodology of the project was empirical in that a manually produced trilingual index was taken as a starting point (12), and routines devised to deal with the problems encountered. New strings were written, designed to test and extend existing procedures. The methods employed owed much to the experience of multilingual thesauri, and where this proved insufficient, then ad hoc methods were used to overcome problems encountered. Techniques drawn from computational linguistics that could have provided elegant and extendible solutions to some of the problems were rejected as "too complex for use with an indexing system" (33).

The system devised centred around three lexicons and a "pivot file", which held a number for each concept together with a pointer to each of the three lexicons. In practice, the information held in the latter was similar to the kind held in a thesaurus rather than a conventional machine translation dictionary. There was for instance, the term and an indication as to whether or not it should be assigned lead status in the manipulation coding. There was no linguistic information, such as syntactic class or semantic information.

The implicit principle of this organisation is that lexical ambiguity must always be eliminated. The majority of PRECIS terms, even taken in isolation, are not lexically ambiguous, simply because of the limited structure of indexing languages (17). Where there was some ambiguity, this was usually overcome by introducing some "concept-setting" term. "Orbits" as used in the BLAISE thesaurus is ambiguous, so it is disambiguated by adding a higher term.

\texttt{az11030fa planets zep1030fa orbits}
The interlingual nature of the system had one disadvantageous consequence in particular. Where a set of terms in one language mapped onto several in a target language, then there was a single pivot file entry that mapped onto the single term in the target, and onto the multiple terms (held in one record) in the source. There is, of course, no way in which the person constructing one lexicon could predict which entries need to be blocked together because it is a factor determined outside the language. In other words the treatment of one language is complicated by characteristics of another - one of the fundamental weaknesses of interlingual systems, and a factor that would make the addition of a new language to the system expensive.

A basic switching procedure was devised, which substituted lexicon entries between the languages, allowing for matching of the longest portion of text. It became clear to the Project team, as it is to anyone who has attempted anything beyond a trivial machine translation system, that prepositions presented problems beyond the capabilities of the simple switching procedure.

Prepositions may occur in three places in a PRECIS string. First, they may be within a term, such as "photographs taken from artificial satellites". In the PRECIS Translingual Project these were already translated in the lexicon. Second, they may appear as connectives - in effect to link the terms together to make the entries from the manipulated strings more natural and unambiguous. Finally, they may appear in substitute phrases, which replace a number of terms in some entries of the manipulated string. Although differently coded in the unmanipulated, they produce formats on the printed page which are identical.

The problem of translating prepositions is that there is rarely a one-to-one correspondence between languages. In other words, the implicit principle of the elimination of lexical ambiguity is breached. In most machine translation systems, syntax is used to resolve lexical ambiguity, and in some, semantic analysis is used to resolve syntactic ambiguity and any residual lexical ambiguity. Ostensibly, the PRECIS Translingual Project had no recourse to such higher levels of analysis, because techniques drawn from computational linguistics had been rejected.

The solution adopted was firstly to separate the processing of connectives and substitutes. As was reported at Information 5 (32), each preposition was given a pivot file number, which (in effect) pointed not to a lexicon entry, but to a flowchart. How these would have been implemented is not clear, but one would speculate that they would have been written as high-level program language code. So the French "de" would be directed via the pivot file to the algorithm for the English "of" (Figure 2). This requires surrounding terms to be examined for their "category number", as well as the role operator originally assigned to them. The category number was a classification of the influence that a word may have on a particular preposition. If a word is normally followed by "of" than (in this algorithm) this represents the
normal translation. Otherwise a particular category number in
association with a particular operator might require "in", "for", "by",
"from", "to" or "as to".

Substitutes were not translated in the PRECIS Translingual Project.
Some substitutes are merely simple noun phrases, and these might be in
lexicons if they were used as terms in their own right elsewhere, though
logically this should never happen. A number have prepositions, which
the Project could only translate by recourse to category number and
operators. As the latter are not present in substitutes (by their very
nature), the method would not work. The alternative proposed was to
automatically create substitutes by using the terms as originally given
in the string and manoeuvring and combining them. This would mean that
the role operators associated with the terms being processed could be
used to translate any prepositions. Another set of categories was
created ("refinement categories") to aid the processing. As time ran
out, the team were investigating the automatic provision of substitutes
in any language, irrespective of whether or not there was to be any
translation.

The algorithms for refining substitutes describe processes of, for
instance, retrieving adjectival forms from the lexicon (an unforeseen
extension to the information that would have to be held); moving the
position of adjectives to before or after the noun, as appropriate, and
enforcing agreement. In other words, carrying out fairly simple,
syntactically based, processes.

The proposed system is difficult to assess with great accuracy, as it
has not been implemented. It seems that it would be a difficult system
to maintain. New strings might introduce new terms which would need
category numbers assigned to them. As these are ad hoc, there would be a
difficulty in maintaining consistency and, perhaps, even passing on
knowledge about how the meaning and interpretation of the categories.
More serious would be the problems involved in the addition of another
language. Putting aside the problems in reviewing all the equivalents in
the lexicons in order to ensure that one-to-many term equivalences are
catered for, there would have to be an analysis of the behaviour of all
prepositions in the new language, and a new set of ad hoc category
numbers created. The seeming willingness of the Project team to write
their linguistic routines as program rather than rules brings to mind
Kay's comments about another, earlier and larger system, that of
Georgetown University: "Such information about the structure of [the
languages being translated] as the program used was built into the very
fabric of the program so that each attempt to modify or enhance the
capabilities of the system was more difficult and more treacherous than
the last" (20).

The use of some of the techniques drawn from computational
linguistics would have undoubtedly slowed the development of the Project
in its early stages, but would it have made any later, positive
contributions? Clearly the answer must be in the affirmative, because as
we have shown above, the PRECIS Translingual Project itself started
introducing some limited syntactic information, although without making
it clear how such information might be applied. Two more advantages
would accrue. Firstly, if an algorithm is devised that will apply the
linguistic information to a string to be processed, this needs to be
proved correct only once (as far as the proof of correctness is possible
with any program). Any changes to the coverage of the system will have
to be made in the rules which the algorithm will interpret, which
compares favourably with the position in which the PRECIS Translingual
Project would find themselves, where the program would have to be tested
again after each enhancement or modification. The use of standard
linguistic theories means that the principles by which the system
operates can be easily communicated. Changes in the performance of
the system may be made by altering the grammar or the dictionary entries.
Obviously, these have to be proved correct, in the same way as a program
has to be tested, but the scale of the problem is considerably smaller.

The design of an alternative system
It was felt that the PRECIS Translingual Project was unsatisfactory in
particular for the reasons given above. A system design was needed that
would allow modification and enhancements to be made easily, and would
lead itself to extension to new languages. Finally, it was felt that a
good system would include some verification of the strings.

The experience of large scale interlingual natural language
translation systems has not been encouraging (18) and even the TITUS
system (10, 11) for the translation of abstracts written using a limited
vocabulary and a restricted syntax has not produced outstanding results.
Bruderer commented of TITUS that the quality was relatively good, but
that the word order occasionally left something to be desired (4). For
this reason, the easier transfer approach was adopted.

The linguistic framework of the system
Clearly the most difficult factor is the translation of prepositions.
Lexical ambiguity is almost completely eliminated at all other points.
As will be shown below, semantic theories offer a reasonable account of
prepositions, but as implemented here, lie on top of syntactic
processing. This was in accord with a separate decision to use a
dictionary of individual words rather than a thesaurus-type lexicon. No
account was taken of inflectional variations, so "observation" and
"observations" had separate dictionary entries, as would "observational"
if it were needed.

Each "raw" string was processed so as to give the longest running
pieces of text for both downward and upward readings (see Figure 3).
Individual pieces of text were taken in turn, so the texts used would be
here:

**Downward**
- Venus
- origins
- theories of Velikovsky, Immanuel
- criticism

**Upward**
- criticism
- theories of Velikovsky, Immanuel of origins of Venus

127
Venus
of
theories
of
Velikovsky, Immanuel
theories of Velikovsky, Immanuel of origins of Venus
criticism
1950-1977
Length of this section : 13
Venus|origins|theories of Velikovsky, Immanuel|criticism| criticism|theories of Velikovsky, Immanuel of origins of Venus| upward text
1 |p |s |3 12 | downward text
1 12 |4 5 7 | downward operators
9 |8 | Length is : 57
2 |2 | upward operators
9 |8 | Length is : 63
Number of dates : 1
1950-1977| d | Length is : 10
10 | Number of o & n operators : 0
Number of lead-only terms : 0

Figure 3
Output from the text preparation module
Each portion was individually submitted to the syntactic analyser, which was of the recursive transition network type, and so produced a phrase-structure tree as a result (Figure 4). Two grammars were used: one for texts introduced by operator "6", and one for all other portions (Figure 5). The analyser was capable of finding all structures for a given text. This happened very infrequently, and resulted in the first-found being accepted, and a message being deposited in the error file to record that an ambiguous string had been found.

One difficulty in explaining the processing of strings relates to how to describe the units of text that are passed to various modules. Clearly, there is nothing like a natural language sentence in PRECIS. The units passed to the syntactic analyser are noun phrases of sorts, but the part of the string passed to the semantic analyser included only those before a "4", "5" or "6" operator (depending on which came first). Here all individual phrases treated separately by the syntactic analyser were considered together.

There was little or no point in performing a semantic analysis of strings that contained no preposition because these would cause little difficulty in translation, so they by-passed the semantic analysis module to go on to the transfer module (16).

Case systems in general linguistics
Case grammar is a distinctive area of linguistics attributable to one man, Charles Fillmore. In 1968 he wrote an article entitled "A case for case", which presented the notion of "deep case", together with some proposals on how this idea could be incorporated into generative grammar. He reasoned that surface case endings show the semantic relations between nouns or noun phrases, and verbs. Moreover, languages without case endings have to resort to other methods (such as the use of prepositions, word order or intonation) to achieve the same effect. He reasoned that underlying all languages must be the same "deeper" structures which are manifested in different ways in the surface structures. He therefore held that case categories must be assigned to the kernel (untransformed) sentences of generative grammar to account for the varying surface forms.

As originally set out, Fillmore envisaged simple sentences in their base forms to include two parts; the modality (which covered features such as tense, negation and mood), and a proposition. The proposition itself consisted of a verb and a set of participants. Each verb had an associated case frame which in effect was a list of the roles that participants were allowed to assume in relation to the verb. Drawing on logic, the verbal element is usually called the "predicate", and the participants the "arguments". "There are only a small number of ways the arguments of a predicate are semantically related to the predicate itself. These ways are called cases" (7).

The verb "hit" may be part of several surface forms; for instance:
"The nail was hit by John"
"John hit the nail with the hammer"
"The hammer hit the nail"
Figure 4
Output from the syntactic analyser
Figure 5
Syntactic analysis grammar
Clearly the reader understands that John was always the agent; that the hammer was the instrument, and that the nail was the patient of the action. A case system is built in a similar way to this, in that a number of verbs are examined and their arguments classified. It is probably true to say that there are as many case systems as there are devisers, although it must be said that a degree of similarity between many systems is easily detectable.

"Selection restrictions" are often placed on the argument "slots" in the case frame, to account for the acceptability of a sentence. "Eat" requires an animate agent, and therefore a sentence like "John ate the apple" is acceptable, whereas "*The post ate the apple" is not. Again there is no general agreement on the selection restrictions to be used, or even as to their names. The ones used in this system were influenced by Chafe (5,6).

Space precludes detailed discussion of the systems described by theoretical linguists, but interested readers are directed to the References (5,8,14,15,24).

Case grammar has been criticised on several points. Firstly, that it is not a grammar as such, for it has no provision for features such as phonology. Hence the term "case system" is preferred here. Second, there is no agreement as to the number and definition of cases. It is theoretically attractive to hope that a definitive list could be constructed (if only because it would be a candidate for inclusion in an interlingua), although this hope seems less bright (25). For practical purposes such as this system, it seems inappropriate to search for a definitive list, in that the need is only for a set that will help to provide an adequate analysis.

Case in natural language processing
A number of natural language processing systems have included a semantic component that may be described as a case system. Depending on how a case system is defined, more or fewer processors may be included in the class. There have been a number of review articles (lamentably none of them recent), of which that by Bruce (3) covers the most. Papers by Sanowski (28), Charniak (7) and Wilks (36) provide an interesting three-sided debate on the use of case in artificial intelligence systems.

Again, space precludes a detailed review of the relevant systems. Two in particular influenced the design of the system presented here. The first was Wilks's English to French translator (34,35); the second was Somers's PTOYS system for "meaning analysis" and "dictionary making", presented at Informatics 5 (29). Wilks's case categories might have been adapted wholesale, but were not mainly because it was felt that categories such as "containment" and "goal" were not necessary for PROCLIS. As Sanowski (28) pointed out, some of Wilks's cases such as "possession" and "accompaniment" were not case relations as Fillmore would have understood them. Somers's system provided the basis of the case relations given below.
The rule base 1 - the dictionary
As has been stated above, the semantic information was superimposed on
the syntactic representation of the string. Therefore semantic
information needed only to be added to lexical items of semantically
significant items. In the current system, it was added to the three
types of noun ("no", "np" and "nn") and the "verbal" element (labelled
"ve") which was used to denote a past participle. For better or for
worse, all other syntactic classes (including adjective) took no part in
semantic analysis. In theory, each dictionary entry could have up to
sixty-three semantic meanings for each syntactic label, although in
practice the actual maximum was three. Predictably, two types of
semantic label was allowed: predicate and argument.

Unlike natural language systems such as PTOYSIS, that use some
syntactic processing to uncover case significant items, it could not be
assumed that only verbal elements would be predicates, even given the
idiosyncratic definition of "verb" used. Actions in PRECIS are usually
in noun form and therefore nouns, as well as "verbs" could be assigned
predicate records.

The predicate record consisted of a number of case slots. For each
case frame there could be one occurrence of each case slot. Also, each
case frame applied to an individual predicate, rather than an entire
string. To recast this in natural language terms, the case frame applied
to the regime of a verb, not to whole "sentences", unlike some case
systems (27).

The cases chosen were:
**Agent** - The usuallyanimate instigator of an action or process:
eg observations by American astronomers

**Experiencer** - The animate entity that undergoes some change of internal
state when instigating an action or process:
eg the hearing of machines by children

**Patient** - The entity on which an action is carried out:
eg the maintenance of satellites

**Factive** - The product of an action:
eg the carving of statues

**Beneficiary** - The animate entity that receives some advantage or
disadvantage from an action:
eg the teaching of mathematics to children

**Instrument** - The usually inanimate entity used to aid a process:
eg the driving of nails with hammers

**Comitative** - Included to allow the expression of two-way relationships:
eg foreign relations of France with Germany
This was not used in the present system, and may be unnecessary for
PRECIS.

**Location** - The locale of an action. It was recognised that this is
usually expressed by the operator "0", but it was anticipated that it
may occasionally occur within a term.

**From location** - The starting point in space of an action. It was
recognised that a temporal starting point may also be assigned this
slot:
eg probes from Earth
To location – The destination usually in space, but perhaps in time, of an action:
  eg space flight to the Moon

Through location – The action of passing through, over or under a locale:
  eg journeys across the Sahara.

For each case slot, three fields of information could be added. Firstly, there were the selection restrictions, which numbered nine:

Biotic
Animals
Human
Physical object
Abstract
Place
Potent
Unique
Predicate

These are on the whole self-explanatory, with the exception of the last. Some slots were typically filled by another case frame. This particularly applied to the instrument slot, which in FIRESHIS frequently takes a form similar to "use of word processors".

Each restriction in a case slot could be coded as having value 0, 1 or 2, where:
  0 – not applicable
  1 – possible, but not mandatory
  2 – mandatory

It was recognised that these nine restrictions might not provide sufficient discrimination to differentiate all cases, so another field was added which could contain an arbitrary number of three character markers. These were envisaged as being similar to the markers used in SYSTRAN (25). These were not needed in this project.

The third field contained the preposition or prepositions most associated with the slot. The rationale behind this was that for the purposes of analysis, certain prepositions would be associated with certain cases, and therefore any residual ambiguity present after the matching of the selection restrictions could sometimes be resolved by the matching of prepositions. If required at synthesis, the appropriate preposition for the case could be taken from the case slot.

The argument record included eight of the selection restrictions given above: biotic, animate, human, physical object, abstract, place, potent, and unique. "Predicate" had no relevance to the meaning of an argument. These restrictions were assigned values in the same way as the restrictions of the case slots, and there was provision for the extra markers described above (again, not needed in this project). In the same way that the dictionary maker could specify that a particular preposition could be expected with a case frame slot, so he could mark that a particular preposition could be expected when the argument filled a certain case slot. This feature was included to facilitate the processing of locational prepositions in French, which seem to be dependent on their argument, rather than their prepositions.
Finally in the description of the dictionary, a note on multiple meanings. As was noted above, some lexical units such as "government" could have both an argument and a predicate record, allowing the two senses present in "The government of Britain" and "The policies of the British government" to be distinguished. The dictionary structure allowed for one or more meanings for an argument, or for a predicate, and for individual case slots. Unfortunately the implementation of the analyser did not recognise these multiple meanings. This presented no problems with the sample used, but might have done so if the sample were extended to the entire BLAISE file.

The programs within the semantic analysis module
With the constraint on the size of individual programs, the semantic analyser had to be split into three parts. The first isolated syntactically important lexical units and retrieved semantic information; the second was the analyser proper; and the final one a module that merged analyses.

The first program processed each syntactic structure of each theme in turn up to a "4", "5" or "6" operator; downward readings first, then upward. Any lexical unit with the label "no", "hp", "nn" or "ve" was stored, together with its attached role operator (if any) and any prepositions or conjunctions before and after the unit (if any). So from a portion of text: "Apollo Project compared with paintings of manned space flight by Smith, Ralph A", "flight" would have "of" and "by" associated with it. The semantic information for lexical units had to be retrieved from the source language dictionary. With upward readings, some limited revision of the order of lexical units had to be done to counter the effect of context-setting terms introduced by the indexer.

The semantic analysis proper was carried out in the second program. This was a two stage process. The first stage determined whether or not a lexical unit should be interpreted as predicate or argument, while the second stage interpreted the results of the first to impose a case frame appropriate to the predicate.

The rule base 2 - the first semantic grammar
The input to this semantic analyser was (in effect) a list of case significant items as described above. The syntactic analysis had been used to remove unnecessary lexical items, and in the process, reducing the number to proportions that would not exceed the size of the machine. The first semantic grammar is shown in Figure 6. It was a network grammar, that tested for the role operators and the semantic category (P or A). Either or both could be optional (see the PRED and PART networks).

This grammar was interpreted using a recursive network parser (RNP). This is a top-down, depth-first searching algorithm that can be made to build its parse tree as the analysis proceeds (i.e. it is more than just a recogniser). In effect it follows paths through networks, making calls to networks where necessary. For instance, the first network requires calls to "FRAM" and "ARGL". Where there are multiple exits from a node, the decision point is saved, so that (in the event of failure to navigate to the end of the network) the parser may return to the most
Figure 6
Semantic analysis grammar 1
recent decision point and explore an alternative. To put it more succinctly, chronological backtracking was built in. Obviously once an analysis had been produced, it could be stored and the backtracking feature used to uncover any alternative analyses.

If no interpretation was found for any reading, this was recorded in the error file and the next reading retrieved (if any). A small piece of program reformatted the results of each semantic structure so that the predicate of each frame was at the beginning of the sequence of items in a case frame. The cognoscente would rightly argue that the use of augmented transition network's structure building registers would be a more appropriate method rather than the use of an FTN and a portion of program. Again, machine size was a limiting factor.

The rule base 3 - the second semantic grammar
The case system used in this project has been described above in detail. The problem that confronts anyone wishing to apply a case system is how to reflect the individuality of a case frame in a general grammar that has to serve all inputs?

Elegant solutions are possible using sophisticated software techniques, but this project was constrained by machine size and therefore had to adopt less elegant techniques.

A small, basic grammar was written (Figure 7), which acted as a pump primer. Associated with some transitions in the network were some primitive actions. These actions were:

LOAD(FRAM) - This was attached to the "pred" test. It stored the current state of the grammar, and then built a new grammar from the case frame of the predicate just subsumed. So if the case frame had the two slots, patient (marked with the selection restrictions, physical object, and predicate), and instrument (with the restriction of predicate), the new grammar would, in part, assume the form:

Here "aP" indicates an argument that is a patient and "PP" and "PI" both indicate calls to other networks. The LOAD(FRAM) action would create the networks:
Here provision has been made for these case slots to be filled by other frames. This clearly shows the recursive nature of the structures envisaged, and also highlights that this project was unable to avoid utterances with more than one predicate.

ERAS(FRAM) - This was attached to the "FRAM" condition. As this was executed after the successful traversal of a subnet that was itself created from a case frame, this action stored the state of the current grammar (ie the one that controlled the processing of the subnet); and the reinstatement of the grammar used to call the subnet.

WRIT(DONE) - This recorded that a particular slot had been filled. It was a necessary feature because (as stated above) the manner in which the case theory was set out allowed only one occurrence of a case for each case frame. Therefore, once filled, it had to be marked as such, together with any variants. So in the example given above, when the patient slot was filled, both the options "ap" and "pp" were marked as filled, and thus could play no further part in the analysis unless they were backtracked over.

A sample "constructed" grammar is given in Figure 8. While the processes above seem clumsy, it should be noted that the information used in constructing the grammar comes from the dictionary itself. Thus in practise, there is separation of the rules and the program that interprets the rules.

An RN was again used as an analyser, with obvious additions to allow the primitive actions (and their inverse for backtracking purposes) to be executed. The matching algorithm was the most complicated of any used in the project. The first stage was to test that the semantic category of predicate or argument was satisfied. If it was an argument then its selection restrictions were matched against those of the case slot. If successful, there was an option to match any prepositions associated with a case against the prepositions attached to the lexical unit. In the version of the analyser used for English, it was the prepositions attached to the case frame slot, although it could have been the prepositions added to the argument record.
Figure 7
Semantic analysis 2
Figure 3
A constructed grammar for "calculation"
The operation of the RIN differed in two respects from the description given above. Firstly, it halted after uncovering the first analysis because space was not available to hold multiple structures for later comparison, although this was what was originally intended. Secondly, if no parse was found, the requirement that prepositions should match was dropped, and the process attempted again.

On success, both the first and second parses were stored for use in the next program. If there were more readings to be processed, then the next was retrieved, otherwise the next program was called.

The final program of the semantic analysis module
If no results were produced by the previous program, then the transfer module of the system was called. Otherwise, each pair of first and second semantic analyses were taken in turn and run together to form a single structure. The various parses for two strings are shown in Figures 9 and 10. The co-ordination of two arguments, \("novae\) and \("supernovae\)\) to fill the patient slot should be noted in the former.

A test sample drawn from BLAISE
The system was implemented on the Digico M28, using the Digico Extended Basic compiler (in lieu of a more appropriate language being available at the time). The core storage available for program and data was about 27k; significantly smaller than many of today's business microcomputers.

The performance of the system was assessed using a sample of strings covering the fields of astronomy and astronautics, numbering four hundred and twenty-three in total. These were taken from the British Library's BLAISE PRECIS DC fiche for May 1961, and more specifically from the ranges 520 to 525.35024550 and 629.4 to 629.8. All but three of these were analysed by the syntactic processor. Of the remainder, only forty-seven needed a semantic analysis. This gave a total of one hundred and ten readings - the discrepancy was due to several strings having two themes.

The output of the semantic module was categorised according to four criteria: where the analysis was correct; where the cases which were represented by a preposition in the string were correctly analysed (although there may be errors elsewhere); where the reading was incorrectly written by the indexer; and where the analysis was incorrect or failed completely.

| Correct Prepositions String Wrong |
|------|------|----|-----|
| 51   | 27   | 7  | 25  |
| 46.4%| 24.5%| 6.4%|22.7%|

The failures were not necessarily carried over to the transfer and generation modules because of the provision of safety-net procedures in these to supply default translations.

An evaluation of the system
It was judged that the case system as described above was adequate to account for PRECIS. The deficiencies of the module can be traced to the
Unification of the first and second parses

FULL

<table>
<thead>
<tr>
<th>FRAM</th>
<th>FRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRED</td>
<td>PRED</td>
</tr>
<tr>
<td></td>
<td>BENE</td>
</tr>
<tr>
<td>pred</td>
<td>pred</td>
</tr>
<tr>
<td></td>
<td>FRAM</td>
</tr>
<tr>
<td>photometry</td>
<td>applications</td>
</tr>
<tr>
<td>PRED</td>
<td>PAT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>pred</td>
<td>FRAM</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>determination</td>
<td></td>
</tr>
<tr>
<td>PRED</td>
<td>PAT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>pred</td>
<td>arg</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>distribution</td>
<td>stars</td>
</tr>
</tbody>
</table>

Figure 10
Output from the semantic analyser - 2
implementation. By far the most serious weakness was the lack of multiple interpretations in the second stage of semantic analysis. This meant in effect that the module was satisfied with an adequate analysis rather than finding a good one. To give an example, "paintings of manned space flight by Smith, Ralph A" was interpreted as "paintings of Ralph Smith's manned space flight" rather than the more obvious interpretation. This deficiency could have been made less serious if the grammar could have been written to allow the most likely parse to be the first accepted, but the automatic construction of the grammar from the case frame ruled this out.

Other weaknesses are less obvious, although with the correction of the above, they would rise to prominence. At several points it has been noted that the more powerful augmented transition network would have been better suited to the task than the RTN used, although space precluded their use. The routine to counteract the introduction of context-setting terms in the first module would be seen to have introduced inaccuracies.

On a higher level, the place of the semantic analyser should be questioned. In the transfer indirect strategy of machine translation it has to come certainly before generation and probably before transfer. What should be examined is how it interacts with syntactic analysis - if it has to at all. This project was tied firmly to syntactic analysis as a prelude to semantic analysis and transfer. In the former, much of the syntactic information was abandoned, and indeed what was used was only used in part. Perhaps a simpler system could be achieved by using multilingual thesauri and dispensing with syntactic analysis completely, but then, of course, it would be very difficult to translate substitute phrases.
References


(13) Ferriger, A-M. Présentation du système d'indexation "PRECIS" d'après l'expérience faite par le Département des Arts du spectacle de la


(22) Lambert, G. PRECIS in a multilingual context: part 4, the application of PRECIS in French. Libri 26 (4), 1976, pp. 302-324.


(33) Verdier, V. Final report of the PRECIS/Translingual Project. British Library Research and Development Department, 1980. (BLRD; 5423).

