THE MONTREAL MT PROTOTYPE[1]

R.I. Kittredge

In a recent demonstration at the Université de Montréal, English sentences covering a wide variety of syntactic types have been successfully translated into acceptable French by computer. Sentences composed using the thousand most frequent English words are entered by an ordinary teletype into the university's CDC 6400 computer which implements a series of transformational grammars and dictionary look-up procedures. Translated output is returned after several seconds delay by means of the same teletype, along with details of certain analysis, transfer, and generation phases. Similar demonstrations using teletypes in Sherbrooke, Ottawa and St. John's have provided translations of sentences by long-distance telephone hook-up to the Montreal computer.

The limited success of this machine translation prototype must be described within *the* perspective of past attempts and persistent problems in the development of an adequate theory of language which would reduce the problem to the technological level. Recent advances in transformational theory have made it possible to relate the "deep" or semantic structures of languages rather than their surface structures. Since its inception in 1965, therefore, the Montreal project has attempted to pass from English to French surface structures by means of deep structures. Such a pivot language *(langage pivot)* was already in use by the group at Grenoble for Russian-French translation. In this approach,

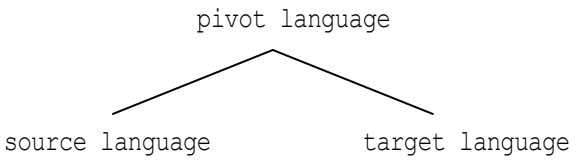


FIGURE 1

structural rules have the effect of converting sequences of words in the source language into an abstract representation in the pivot lan-

guage, from which another set of rules produce a sequence of words of the target language (see fig. 1 above).

The difficulties in using a pivot language are largely practical. When a single tree structure is to serve as the representation of both a French sentence and the corresponding English one, it is necessary to posit abstract lexical entities which are realized differently in the two languages. A more immediate consideration has to do with the co-ordination of work in developing an analysis grammar for the source language and a synthesis grammar for the target language. The work is made as independent as possible when the two grammars are separated by a distinct transfer stage which makes lexical substitutions and small structural adjustments. The strategy taken by the Montreal group can then be represented (see fig. 2) as the passage from English to "normalized" English followed by transfer to normalized French, followed by synthesis (generation) of French.

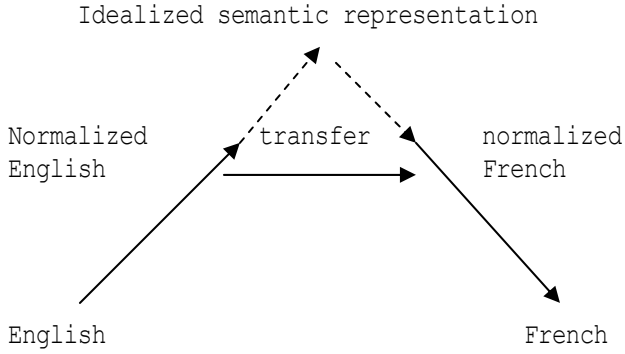


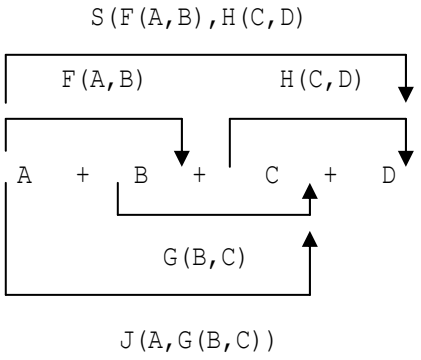
FIGURE 2

A further advantage of this procedure is that the two language grammars are more readily usable when paired with other languages. In the absence of a complete theory of semantic universals, the choice of a pivot language is in practice language-pair-specific. This is not necessarily the case for the passage through normalized structures. By replacing only dictionary rules (some of which will depend on structural properties) in the transfer stage one could match the English analysis half to a Russian synthesis half.

2. Q-systems and transformational grammar

The POLYGRAM translation program in its present version consists of twenty-one consecutively-executed grammars, each written in the Q-system formalism.[2] In simplified terms, a Q-system is a set of rules, each of which rewrites a sequence of trees. Rules may contain variables which stand for labels, trees, or embedded sequences of trees. Rules may have complex conditions on them involving the inclusion of one tree or tree variable in another.

The reading and execution programs cause a Q-system grammar to operate on an input sequence of trees roughly as follows. All possible sequences of rules of the grammar are applied to each substring of the input string or to strings which result from application of rules to such substrings. The output from the grammar is the resultant of the longest derivation(s) through the rules. A simple tree-building example is given in fig. 3 below.



(1) A+B == F(A,B)
(2) B+C == G(B,C)
(3) C+D == H(C,D)
(4) A+G(X*) == J(A,G(X*))
(5) F(U*)+H(V*) == S(F(U*),H(V*))

FIGURE 3

The symbols U*, V* and X* are variables for embedded strings of trees. As can be seen, non-embedded trees are separated by the plus sign from each other. Embedded trees are separated by commas. The input string A+B+C+D under-goes the operation of rules (1), (3), (5) as well as the sequence (2), (4). A precedence mechanism, however, chooses the first sequence over the

second since it includes as its resultant more of the input string than the second rule sequence. Thus for the Q-system consisting of rules (1)-(5), the output is the sequence consisting of the single tree:

S(F(A,B),H(C,D))

For a grammar consisting of only the rules (1)-(4), however, two resultants would be given since neither resultant consists of trees which include the trees of the other resultant. Thus there are the two output strings:

J(A,G(B,C))+D

and

F(A,B)+H(C,D)

For a grammar containing the rules (1)-(5) plus the additional rule (6) J(U*)+D=S(J(U*),D), the output would be the two structures:

S(J(A,G(B,C)),D)

S(F(A,B),H(C,D))

Such multiple output occurs in the syntactic analysis grammar of English. A sentence which is structurally ambiguous may have two or more such rule paths and hence lead to multiple structural interpretations as in this last example. Each such structure is then passed on through the subsequent grammars, giving rise to the same number of French sentences as final output.

The grammars used for analysis function much like the example of fig. 3, i.e., their rules are highly interactive (though unordered) and tend to produce a single complex tree out of simpler trees. (For the synthesis grammars the opposite is the case.) For the dictionary phases, most rules simply replace a label (degenerate tree) by a simple tree, as for example:

BANANA == N(BANANA,/,CONCR).

The twenty-one Q-system grammars which make up the POLYGRAM translation program are executed in sequence. That is, input to the first grammar is an English sentence and the output of the *nth* grammar serves as input to the n+1*th* grammar ($1 \leq n \leq 20$). Output of the twenty-first grammar is normally a French sentence. In terms of function,

this sequence can be divided into seven major consecutive sections: (a) recognition of constants (e.g., prepositions, conjunctions, irregular past, participal and plural forms) idioms which have no substitution positions are recognized here also. (b) decomposition of words not recognized by preceding sections into letters for recognition of important prefixes and inflectional suffixes. This is followed by recomposition and restoration of the base to the uninflected form. (c) dictionary for uninflected forms (essentially all nouns, verbs, adjectives are identified and assigned syntactic/semantic features here). (d) structural analysis of English using the syntactic categories provided by the preceding phases. (e) transfer stage: replacement of English lexical items by French in the tree structure, (f) generation of French surface structure and (g) French morpho-graphemics (including conjugation of all verbs, elision, etc.).

In stages (d) and (f) the Q-system rules often function as transformational rules do. For example, in the English analysis section, the rule:

```
IT+T(A*)+BE+NP(U*)+REL == NP(U*,STRESS)
```

is recognizable as one which converts certain cleft sentences (it-extractions) into normal sentence form while preserving a feature of stress on the topicalized noun phrase. Thus in processing the sentence: *She said it was John who came late,* the structure is normalized to one corresponding to: *She said John came late* where the feature of stress within the list of features for *John* is preserved through the transfer stage so that an embedded cleft sentence may be produced in French. The motivation for carrying oat this regularization of structure is clear when one considers the problems of inducing an ordering (by means of label changing) for the syntactic recognition rules.

The flexibility of the Q-systems comes in the fact that it is quite easy to modify the grammar by changing a single rule. The reading and execution programs are not affected since they are designed to interpret and execute an arbitrary Q-system grammar. These changes can be carried out by instructing an up-date program through the teletype terminal. Changes in grammars (including dictionaries) as well as tests of sentences for translation can therefore be

executed from any teletype by placing a long distance call *to* the Montreal computer's system TELUM.

3. The morphological phases (sections a, b, c)

Extensive grammars for the recognition of English morphology have been developed by using a feature that permits the decomposition of a word into individual characters. Such an approach has eliminated the necessity of having a full-form dictionary, since rules can be written (within the Q-systems) for recognition of important prefixes and the set of suffixes *-s, -ed, -ing, -ly, -er, -est.* Prior to this stage, constants (e.g. prepositions, conjunctions, pronouns, modals, etc.) have been recognized and assigned to grammatical categories or given temporary labels to block the decomposition process on these forms. Similarly, irregular verb forms and irregular plurals have been identified by consulting a virtually exhaustive listing. Since these forms are immediately given category markers, they are unaffected by the suffix-splitting phase which splits only simple labels. The splitting phase then separates prefixes from both regular and irregular forms, and suffixes from regular ones. After affixes have been removed in the splitting phase, a complex set of rules adjusts the spelling of the root to that of the uninflected form (e.g. *flie* + *s becomes fly* + s). Dictionary search follows and for forms such as *fly* all possible lexical interpretations, N(FLY,...), V(FLY,...), etc., are recorded for that position of the input string. The interpretation of the suffix *-s* can then be made dependent in the following grammars on the category markers found, by rules such as:

```
N(U*)+s == N(U*,PL)
V(U*)+s == T(PRS3S)+V(U*)
```

In the case of *flies,* both rules apply and both nominal and verbal interpretations are available for subsequent rules. It is, of course, necessary to search the dictionary for the full form as well as the split form. Entries such as *trousers* will be split into *trouser+s,* but since *trouser* is not listed in the dictionary, the program has the effect of giving the entry for *trousers* which has been looked up at the same time.

Dictionary rules assign most words to one or

more grammatical categories by rewriting the word as the left-most branch of a tree dominated by the appropriate category label. At the same time certain syntacto-semantic features are listed as remaining branches. Some sample listings from the open dictionary classes (nouns, verbs) adjectives) are as follows:

```
          OIL == N(OIL,/,CONCR,MSS).

LOVE == ZV(LOVE,/,1(HUM),2(HUM,CONCR,ABST),
                  STAT).

           TRUE == ADJ(TRUE,/,S)
```

Words not belonging to the open classes, essentially transformational constants, have been recognized by rules in the initial phase and have been assigned category labels but without features. Among these constants are included prepositions, conjunctions, quantifiers and articles. One additional class recognized in the initial phase is that of so-called "circumstantials", mainly those adverbials not derivable by general rules from adjectives.

4. Syntactic recognition phase (section d)

After identification of constants and irregular forms, followed by dictionary look-up, the output of the morphological phase is passed into the first syntactic recognition grammar which builds a preliminary NP-dominated tree by recognizing pre-nominal modifiers and assigning them to a standard position in the tree according to the following schema:

```
     NP (<noun><determiners><adjectives>/
             <noun features>).
```

The two strings <determiners> and <adjectives> may be empty or may be built up in a right-branching fashion from conjunctions of DET-dominated or ADJ-dominated strings. Certain local modifiers are converted into the features on their head elements. Comparative forms of adjectives, for examples, are recognized by the two following rules:

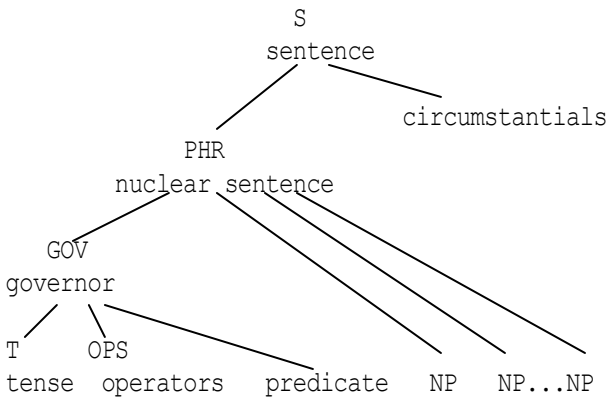
```
       MORE+ADJ(U*) == ADJ(U*,COMP).

        ADJ(U*)+ER == ADJ(U*,COMP).
```

Similarly, superlative forms and pre-adjectival *very* are converted to adjective features.

Following preliminary NP buildup, the second syntactic grammar carries out the process of sentence recognition. Although the rules are unordered by the system, an ordering is effectively induced by the changing of tree labels as more complex constituents are built up. The initial step is the recognition of the governor or predicate which carries the tense. This may be a verb with auxiliary, or the *be* copula with adjective, classifier noun, or circumstantial. The governor constituent then, dominated by GOV, consists of three parts in the canonical order: GOV(T(A*), OPS(U*), <pred>) where <pred> is of the form V(V*), ADJ(V*), NP(V*) or CIRC(V*).

The rules which recognize the canonical order of sentence elements depend on the basic string property of sentence being: $Noun_1$ + Aux + Verb + $Noun_2$ + $Prep_1$ + $Noun_3$ + ... The nuclear sentence is built up by recognizing a constituent noun phrase (NP) preceding the Aux+Verb element (GOV) plus any following noun group or preposition+noun group. Thus, in terms of a predicate/argument view, the argument NP's are recognized and grouped with the predicate one at a time. Circumstantials which interrupt or adjoin the sentential nuclear string are incorporated into a sentential tree by adding them under the sentential label S as a branch on the same level as the nuclear sentence PHR. This gives an overall representation of the sentence as follows:



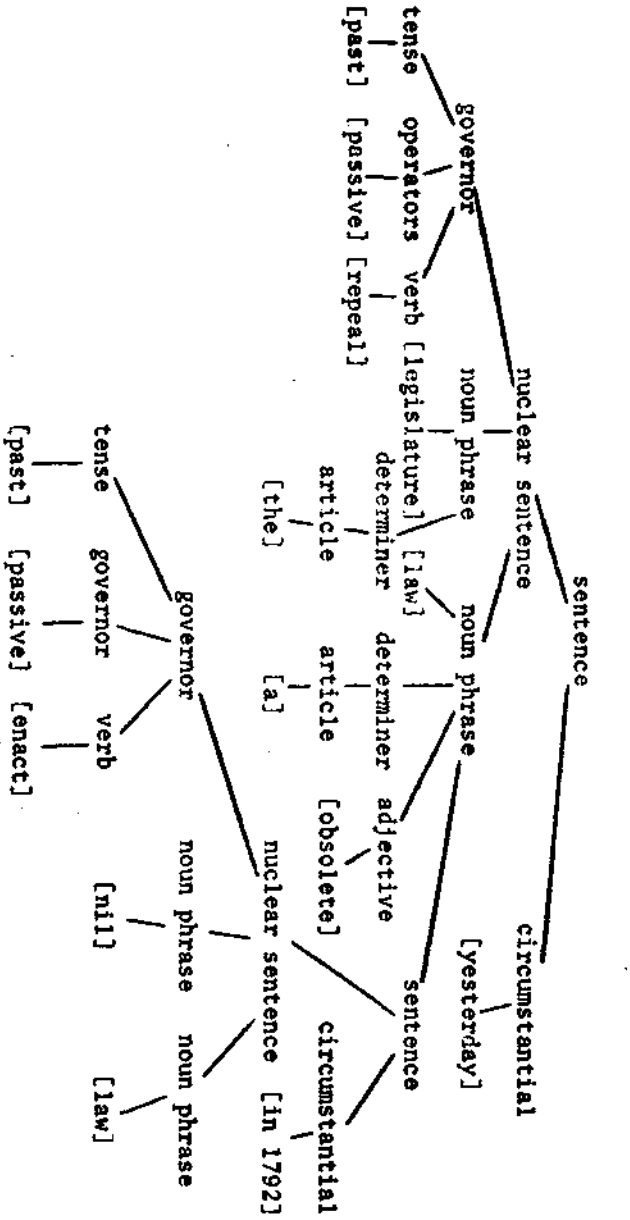
When any constituent (sentence, NP, adjective,

circumstantial) is conjoined to another of the same category with *and, or* or *but,* the structure of the conjoined pair is:

<label>(<conjunction>,<constituent$_1$ >,
<constituent$_2$ >,/,<common features>)

For parts of sentences which interrupt the canonical order, special rules must be given. Thus passive forms, relative clauses, nominalizations, etc. must be recognized by using the transformational trace (e.g. *be...-en}* and the grammatical rules reconstitute the canonical form with the addition of the information (e.g. PSV under OPS) that triggers the generation of the corresponding non-canonical form in French.

A complete description of the normalized form for sentences is usually given by a set of Backus normal statements. A single example here will perhaps give a clearer picture. The sentence *An obsolete law which was enacted in 1792 was repealed by the legislature yesterday* has the following tree structure (omitting details)



In noun phrases with relative clauses, the relative clause is reconstructed as a full sentence with the relativized noun phrase repeated in it. For passive sentences, the agent noun phrase occupies the position of logical subject by virtue of a set of rules which function when the PSV (passive) marker is present in the governor.

When no agent is present (e.g. t*he law was enacted)* the logical subject position is filled by a dummy: *nil.*

5. Transfer phase (section e)

During the transfer phase, the tree output of the English syntax is segmented in such a way that branching information is preserved, but each English lexeme is put at the highest level. Essentially, for the Q-system purposes, each sentential tree becomes a sequence of trees. This is necessary so that a Q-system grammar can be written which contains single lexical transfer rules. Many rules are of the simple replacement type: BOOK==LIVRE. They are often made dependent on their syntactic environment (as represented in the tree structure). Thus it is possible to have KNOW==SAVOIR under the condition that the second (object) noun phrase consists of a sentence (or under a few other syntactically specifiable conditions). Under remaining conditions the -rule KNOW==CONNAITRE will apply.

When dictionary transfer rules have finished applying, the tree is recomposed (in a subsequent Q-grammar) to give a normalized structure for the French sentence. Besides lexical transfer, however, certain small changes in category labels and tense markers have been carried out. When the past tense marker is found with the BE marker (for the progressive *be... -ing)* a rule produces the marker for imparfait in French. The output of the transfer stage, then, produces a tree which has essentially the same structure as the English normalized structure tree. The limitations of structural transfer are obvious for many translation problems. Certain correspondences can be expressed structurally with the help of features (e.g. *stative,* etc. for verbs, *group, body,* etc. for nouns) which identify important syntactic/semantic word classes. In the absence of a comprehensive semantic theory, the limitations of structural transfer will have to be accepted as the best we can do.[3]

6. Generation of French surface structures (section f)

Each recomposed tree-structure which enters the French generation grammar presumably corresponds to a single interpretation (non-ambiguous) of an English sentence. Although many French paraphrases could be generated from this structure,

the rules for French generation produce only a single sentence by testing markers in various components of the sentence. Sentence conjunction, question or passive markers trigger rules which affect the global unravelling of the tree. Relative clause structures undergo a rule which effectively permutes the duplicated noun phrase and pronominalizes it to produce the surface relative *qui,* etc. Rules which test object noun phrases for pronouns perform permutation of these pronouns to the position before the verb. Other "local" rules write *très* or *plus* before adjectives or certain circumstantials on the basis of markers found in the ADJ or CIRC tree.

What results from the structural manipulations is a string of words with additional markers (for gender agreement, for example) to be used as input to a multi-stage morphology program (section g) which essentially generates the proper verb form and carries out gender and number agreement on adjectives. Other temporary markers which have served in the derivational processes are erased at the same time.

7. Prospects and limitations

The Montreal MT prototype has a certain flexibility which permits rapid implementation of new developments in transformational grammar. It is easy to add or remove a grammar from the entire sequence. Likewise it is easy to insert or delete a single rule or group of rules, generally without requiring serious adjustments in rules that remain. Given this flexibility, it is possible to foresee a rapid expansion of the English lexicon and transfer dictionary well beyond their current 1000-entry size. Grammatical segments may also be expanded to include most of the well-established facts about the behavior of English and French syntax. Within a very few years, however, the limitations of structural transfer will become the main hindrance to improvement of quality, so that semantic theory and especially descriptions of discourse structure must be developed apace if high-quality automatic translation for extended texts is to be achieved.

NOTES

1. This paper describes the translation system of the Groupe de recherches pour la traduction automatique à l'Université de Montréal. A full

listing of the component grammars and dictionaries is available in TAUM 71 a report published by the group in January, 1971. At the time of publication, the project's principal members were: Linda Anderson, Alain Colmerauer, Jules Dansereau, Brian Harris, Richard Kittredge, Guy Poulin, Gilles Stewart, François Stehlin, and Michel Van Caneghem. Copies of the report in microfiche and spiral-bound paper are available from project secretary, Louise Valiquette, T.A.U.M., Université de Montréal, Case Postale 6128, Montréal (101), Qué.

2. A fuller description of the Q-system, designed by Alain Colmerauer, is available in the TAUM 71 report described above.

3. Since the implementation of the version described here, in February 1971, certain proposals for semantic representation and disambiguation have been made by R. Hofmann and others. Some of these are being developed as an additional phase following the English syntactic analysis phase.

Richard I. Kittredge
T.A.U.M.
Université de Montréal
Case Postale 6128
Montréal (101), Qué.

September 1971.