[From: *Perspectives in artificial intelligence, vol II*, ed. J.A.Campbell and J.Cuena (Chichester : Ellis Horwood, 1989)]

EUROTRA: the machine translation project of the European Communities

B. Maegaard
University of Copenhagen, EUROTRA-DK, Njalsgade 80, DK-2300 Copenhagen S, Denmark

1. INTRODUCTION

The EUROTRA project is the machine translation programme of the European Community. In 1982 it was decided by the Council to implement this programme, but only by late 1984 were the first contracts between the Commission of the European Communities and some countries signed. The goal of the project is to develop a pre-industrial prototype for machine translation between the nine community languages. When the decision on EUROTRA was taken there were only seven languages, but with the accession of Spain and Portugal last year we now have nine languages. The prototype should be ready in 1990.

The Council decision further states that the prototype shall work for a vocabulary of 20000 lexical entries, for a limited subject-field and for a limited set of text types. The subject-field is not determined by the Council decision; it has been chosen to be information technology (IT). The set of text types has not been fully defined yet; the text types in question will be Commission texts, such as Council decisions, working papers, etc.

Apart from this the Council Decision of 1982 requests that the prototype be extensible: it must be possible to extend the coverage of the vocabulary to other subject fields, to extend to other languages, and to extend to other text types.

The components of the system are being developed by all the Community countries and the project is managed by the Commission in Luxembourg. So, we do not only have the task of building a machine translation system. There are two very important additional factors which have to be taken into account.

First we are faced with a very high degree of decentralization with 12 countries and the Commission, i.e. 13 participants. Furthermore, in some countries the work is further decentralized in that the EUROTRA group is made up of two or more centres. The system design has to take this into account.

Secondly, the programme is multilingual, not bilingual or just comprising a few language pairs. This project is unique in that it comprises 72 language pairs. What this means for the linguistic descriptions is considered later.

2. DESIGN

EUROTRA uses a variant of the transfer model for machine translation, i.e. the translation process is broken down into three modules, analysis, transfer, generation:

text ———---- IS -----— IS' ------------ text'
analysis transfer generation

This is generally acknowledged to be a very good scheme for multilingual machine translation, as it restricts the bilingual treatments to the transfer modules: only one analysis module is made for each language, and only one generation module. There will be transfer modules for all language pairs, i.e. 9 × 8=72 transfer modules in our case. (Of course an even better scheme in an environment which is multilingual to this extent would be a transfer-free, i.e. fully interlingual, approach. For the time being, however, this is not a practical possibility.)

The monolingual components are made in the various countries, Danish in Denmark etc., and the transfer components are made in collaboration between two language groups, with the target group as mainly responsible.
In the EUROTRA framework we have generalized the transfer model

text→R1→R2→… .→Rn ————Rn'→… .→R2'→R1'→text'
transfer

The mapping Rn→Rn' is the original transfer mapping.

We are working with

(1) a base level, which will probably be broken down into more levels of representation, EBL;
(2) a constituent structure level, ECS;
(3) a syntactic relations level, ERS;
(4) a semantic relations level, IS.

Each level of representation is defined by what we call a *generator,* i.e. a grammar and a dictionary. The mapping between levels is performed by a *translator.*

A generator consists of *structure-building rules* and *non-structure building rules.* Structure-building rules are context-free rules operating over objects which are *feature bundles.* The context-free rules not only refer to categories (such as N, NP etc.) but may also mention features in the feature bundles in question. Most of the feature manipulation is done by the non-structure building rules, however.

Here is a short example to show how the generators and translators are supposed to work. The example is made according to a definition of the EUROTRA framework which was used in the spring of 1987.

Let us consider an ECS rule for a noun phase:

Structure-building rule

```
            np2 = (np)
            [ ^(detp)
            *(adjp)
            (n, {case=ngen})
            ^(pp)]
```

This ECS rule will build a noun phrase out of an (optional) determiner, zero or more adjective phrases, a noun, and an (optional) prepositional phrase. The annotations to this structure-building rule contain the following rules:

Non-structure building rules

```
killer:
          aknp1 = (np)
                     [?,*,(n,{def=df}),*]
strict:
          asnpl = (np)
                    [ ^(detp, {gend=G, numb=N, def=D}),
                    *(adjp, {gend=G, numb=N, def=D}),
                    (n, {gend=G, numb=N, def=D}),
                     *]
gentle:
          agnpl = (np, {gend=G, numb=N, def=D})
                    [*,
                    (n, {gend=G,m numb=N, def=D}),
                    *]
```

These non-structure building rules, or feature rules, work as follows.

The killer rules will delete a structure built by the structure-building rules, if they unify, i.e. the aknp1 rule will delete an np-structure, if the noun of the np is definite. The example here is taken from a Danish grammar. For example, in Danish we may have noun phrases such as

	English translation:
forslag	proposal
forslaget	the proposal
det bedste forslag	the best proposal

The strict rules, like killer rules, can delete structures that have been built. Strict rules are typically used to check agreement: the structure will be deleted if the components do not obey the rules expressed by the strict grammar rule. In the actual case of a Danish noun phrase, agreement is required between the (optional) determiner, the (optional) adjective(s), and the noun.

Finally, the gentle rules are used for percolation etc. They do not delete anything; they only add information. In the actual case of the Danish agnp1 rule, it percolates the values of gender, number and definiteness from the noun to the resulting noun phrase.

At ERS level the corresponding rule could be

```
np3 = (-, {cat=np}
        [(gov, {cat=n})
        (^ mod, {cat=detp})
        (*mod, {cat = adjp})
        (^ mod, {cat = pp})]
```

A t-rule which translates an ECS structure built by the np2 rule into the corresponding structure at the ERS level could then be

```
tnp10 = (np)
         [$B(^ detp),$C(*adjp),
          $D(n),$E(^ pp)]
          → np3($D,$B,$C,$E)
```

In this scheme all nodes have to be translated explicitly, and furthermore it is already decided by the t-rule what structure-building rule to apply at the next level (np3 in the above case).

We see this as a problem when we get to bigger systems and more complicated structures. Therefore a proposal has been made for a slightly modified system where the t-component becomes a little weaker, and in particular where the generator has more power.

Basic ideas about *generators* and *translators* are now considered. Generators are context-free rules with annotations, as described above. Translators are (1) one-shot and (2) compositional.

That translators are ‘one-shot’ means that they map from one level of description directly to the next level of description, i.e. there can be no intermediate representation (such a representation could not be checked for wellformedness).

The basic objective of ‘compositionality’ is that the overall image of sentence, on translation, can be obtained from the images of its parts.

Now, if translators were totally compositional, they would be homographies in the mathematical sense, and t-rules such as the one above, which manipulates the order of constituents, would not be allowed. Consequently we are using a relaxed version of compositionality, where it is possible to change precedence between sister nodes, to change dominance, to delete nodes, and to insert nodes.

In the course of spring 1987, work on a slightly different version of the same ideas has been going on. It has resulted in a new prototype which will be used for implementation at least until the end of the second phase of the project.

The main difference of the new approach is exactly that the nature of the translators is changed: as we can see in the earlier framework, the t-rule determined the structure to be built at the next level. In the new framework this is not the case; the t-rules will in general be weaker than before, whereas the generators will have more expressive power than before. This is a sound principle as it makes the generators more autonomous, and makes the implementations more easily modularizable.

The main principle is that translators deliver as input to the next level a set of nodes, with 'soft' precedence and dominance relations between the nodes. What can be done to this 'softly structured' object by the generator, apart from just consolidating the structure, is that nodes can be inserted both in the horizontal and the vertical dimension. Thus, for example, if

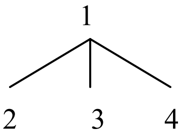

Fig. 1.

is the input from the previous level, then the shape of the object may be one of the following after application of the generator:

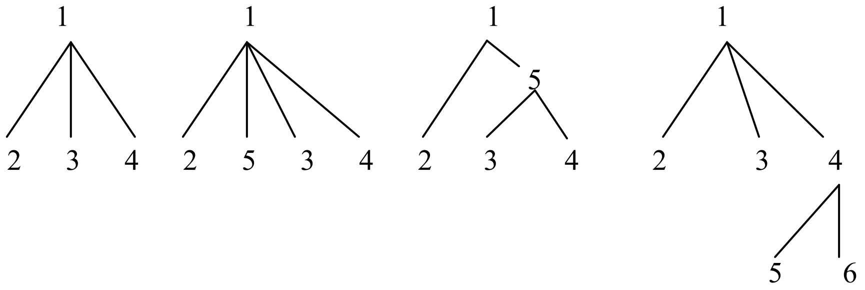

Fig. 2.

3. THE DEFINITION OF THE LEVELS

The interface structure which is the common exchange format has to be totally defined, and all language groups have to use the same definition.

For the other levels, however, this is not necessarily the case. The closer we get to surface text, e.g., the more divergent the descriptions may be. We do try to have a reasonably well-defined set of representations or set of ideas of representations that is used for all languages. This is an advantage for the communication between groups, but it is in a way not a necessary condition; only a common definition of the interface structure is necessary. Having common ideas about the lower levels as well provides a common basis of understanding for the various language groups, however, and we should therefore allow only for as much freedom as is necessary for handling the various languages adequately.

Here is a brief example of a translation of a simple sentence from Danish ECS to German IS. As can be seen (Fig. 3) the ECS structure reflects the surface word order. The Danish input sentence is ‘I 1982 blev alle forslagene vedtaget af Komissionen’. At ECS the constituents are built.

At ERS level the surface order is abolished: ERS and IS both have a fixed order of constituents. At ERS the surface syntactic relations are determined (Fig. 4).

Then finally at IS level (Figs. 5 and 6), the case roles of the various constituents are determined. Surface phenomena such as argument-bound prepositions, determiners, etc., disappear structurally at the IS level—they are expressed by other means.

The case-role system which is used is very simple. It is complicated to define case roles (like ARG1, ARG2, ...) in a way that accounts for aft languages; this is why for the time being we are using this very simple set of roles, which can then be supplemented by lexical semantic features. As can be seen from Fig. 6 the German IS is very similar to the Danish IS. The translation process continues from German IS to German ERS, ECS and text. (This is not shown in the figures.)

4. MORE COMMENTS ON THE IS LEVEL

For the distinction between ambiguous words, lexical features of the semantic kind are needed. For example, we require features of the type human-non-human, concrete-abstract, and all the subject-field features known from ordinary dictionaries, such as zoological, medical, etc. Here we should remember, however, that for the time being the project is working within the subject-field of information technology and distinctions involving other subject-fields, too far away, are not taken into account.

Consequently what is taken into account are word-senses that fall within the subject-field of IT and neighbouring fields, as well as the general senses. Neighbouring fields in Community terms are administration, economy, legal aspects, ...

In order to make transfer simple we disambiguate monolingually as much as is reasonable. What is reasonable can be seen from the distinguishing features. If a lexical unit can be distinguished by e.g. frame, it splits according to the semantic features of one of its constituents etc. There are words, however, that do not lend themselves in a reasonable way to a monolingual disambiguation. There are three possibilities:

(1) The disambiguation is done in transfer, i.e. with access to the two languages involved.
(2) The disambiguation is done in generation, by the target language generator and dictionary.
(3) This will be done in the eventual analysis, and will be decided in negotiation between the two language groups.

It should be stressed that the monolingual solution, i.e. either analysis or

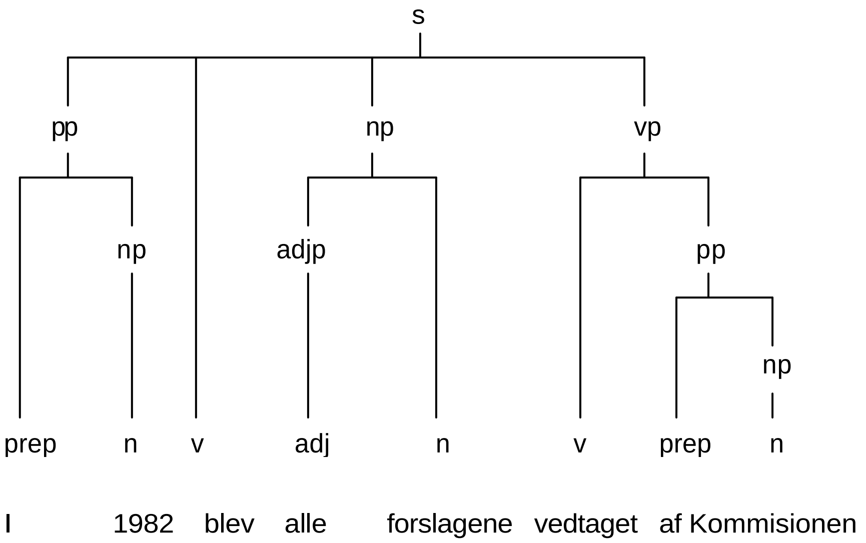

Fig. 3.

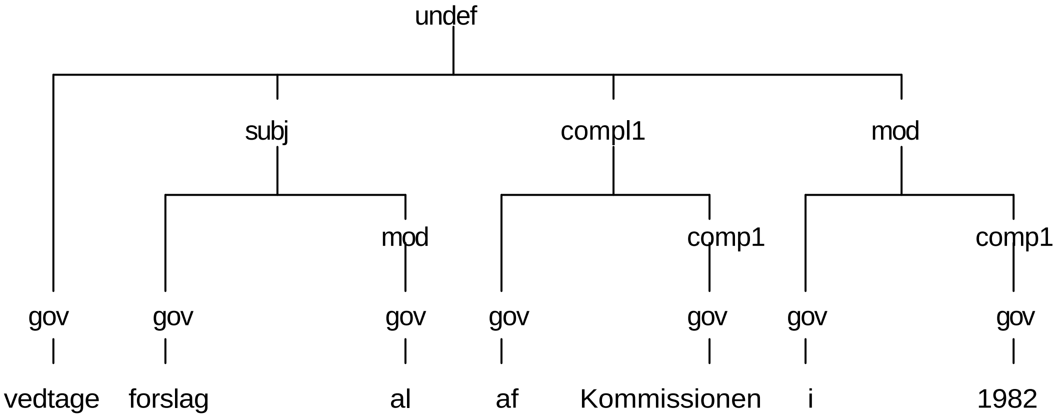

Fig. 4.

generation, is preferable, because the transfer component should be kept as simple as possible.

One of the problems in the lexical transfer is in fact what a lexical unit is: what is the unit which we want to translate and consequently which we want to list in our dictionaries?

Here the opinions in EUROTRA are quite divergent: some people would like to do the translation in the most elegant way, that in our case would be to split everything into small units which could be translated by simple transfer and then recombined by the target-language grammar in a

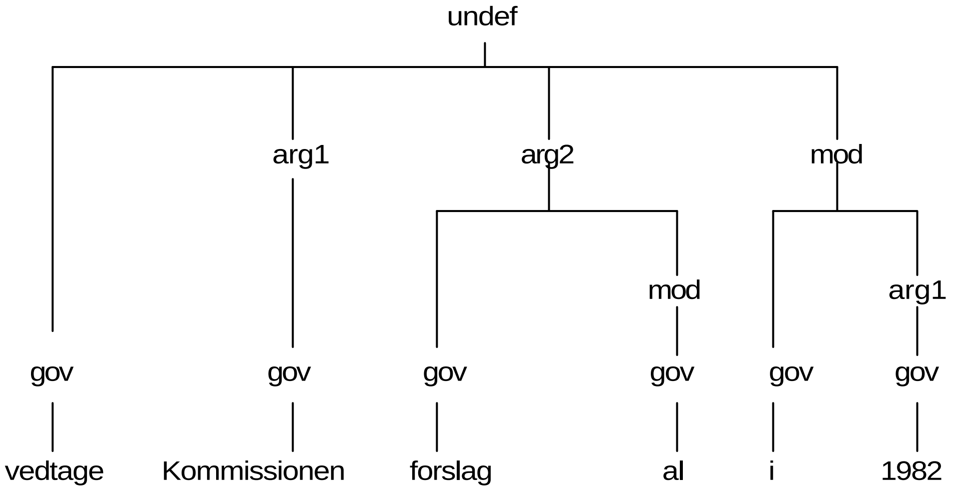

Fig. 5.

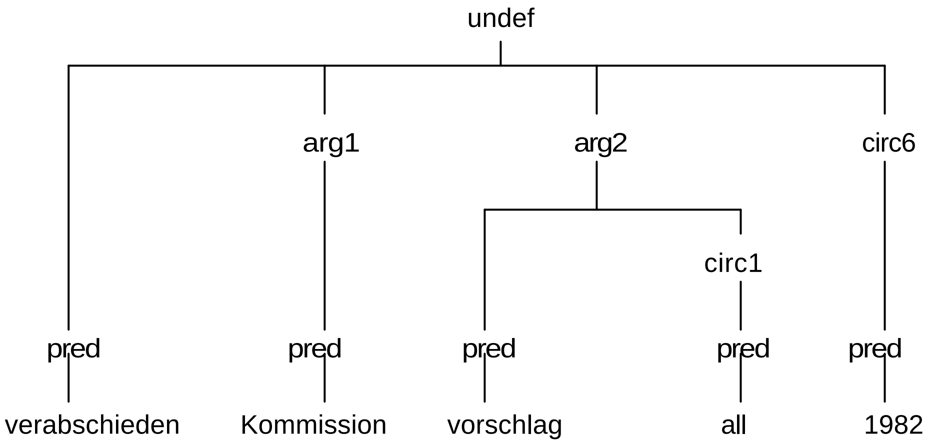

Fig. 6.

correct way. This is possible only with an interface structure which has a very high degree of interlinguality.

Consequently other Eurotrians find that the safest approach is to put bigger parts of the text into the dictionary, e.g. derivations and compounds, in so far as they are 'lexicalized' and are of course idiomatic expressions and terms. In fact nobody argues about the idioms and the terms, but it is not so easy to see when a compound is lexicalized.

For the time being we are not splitting derivations and compounds into their parts; in the future, if a good method comes up, we may do it.

One of the reasons why compounds come up as a problem is of course that Danish, German and Dutch have a compounding mechanism whereby words are glued together to form one single string. However, this is not the heart of the problem; the problem is the 'context-sensitive' translation and how to handle it. Take as an example (Danish-French).

handelsoverskud	excédent commercial
handelsminister	ministre du commerce

5. THE STATUS OF THE PROJECT, SUMMER 1987

The Council Decision of 1982 divides the project period into three phases. The first phase is preparatory; the goal of the second phase is to develop the theory of translation and to implement it in a small machine-translation system which covers all languages with a vocabulary of 2500 lexical entries. The second phase finishes in July 1988. The goal of the third phase is to extend the small system of the second phase and to cover a vocabulary of 20 000 lexical entries.

In February 1987 the first small-scale translation system was finished. It had a vocabulary of 500 words, grammars only for simple sentences, and it worked for translation between German, English and Danish. Since then the coverage has been extended, in terms of language pairs, in terms of vocabulary and in terms of grammar, and we believe that at least for the languages which were part of the programme from the beginning good results can be obtained by July 1988 — special programmes have been initiated for the Spanish and Portuguese languages, as these became part of the project only recently.

EDITORS' NOTE

We have included this survey in the current volume because of the potential interest of the EUROTRA project for natural-language specialists in AI. The scope of the project, especially with respect to the large number of language pairs involved, suggests that there will be many specific problems that can be attacked usefully by AI methods when the more conventional methods used in traditional automatic translations are not sufficient. The chapter serves partly to indicate what the present EUROTRA approach is, but mainly to spread information about the existence of EUROTRA and its goals more widely among the AI community.