Eurotra: The Machine Translation Project of the European Communities

B. MAEGAARD
University of Copenhagen, Denmark

Abstract

This paper describes EUROTRA, focusing especially on two aspects: the general formal framework for the translation, and the dictionaries.

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 late 1984 the first contracts between the Commission of the European Communities and some countries were signed. The goal of the project is to develop a pre-industrial prototype for machine translation between the 9 community languages. When the decision on EUROTRA was taken there were only 7 languages, but with the accession of Spain and Portugal last year we now have 9 languages. The prototype shall be ready in 1990.

The Council Decision further states that the prototype shall work for a vocabulary of 20,000 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, like 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, not only do we 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 decentralisation with 12 countries and the Commission, that is 13 participants. Furthermore, in some countries the work is further decentralised 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 *multi*lingual, not bilingual or just comprising a few language pairs. This project is unique in that it comprises 72 language pairs. I will get back to what this means for the linguistic descriptions.

Design

Having presented the goal and organisation of EUROTRA very briefly, I will now first discuss the system design. EUROTRA uses a variant of the transfer model for machine translation. I.e. the translation process is broken down into 3 modules: Analysis, Transfer, Generation.

$$\text{text} \xrightarrow[\textit{analysis}]{} \text{IS} \xrightarrow[\textit{transfer}]{} \text{IS'} \xrightarrow[\textit{generation}]{} \text{text'}$$

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 main responsible.

In the EUROTRA framework we have generalised the transfer model

$$\text{text} \rightarrow \text{R1} \rightarrow \text{R2} \rightarrow \ldots \rightarrow \text{Rn} \xrightarrow[\text{transfer}]{} \text{Rn'} \rightarrow \ldots \rightarrow \text{R2'} \rightarrow \text{R1'} \rightarrow \text{text'}$$

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

We are working with the following levels of representation:

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*.

I will now first describe a generator. 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 do not only refer to categories like N, NP etc.' but may also mention features in the feature bundles in

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

question. Most of the feature manipulation is done by the non-structure building rules however.

We will here give 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 phrase:

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 = dn} ),*]
strict:
      asnp1 = (np)
              [∧(detp, {gend = G,numb = N,def = D}),
              *(adjp, {gend = G,numb = N,def = D}),
              (n, {gend = G,numb = N,def = D}),
              *]
gentle:
      agnp1 = (np, {gend = G,numb = N,def = D} )
              [*,
              (n, {gend = G,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, because such a noun cannot combine with a determiner or adjective.

(This example is taken from a Danish grammar. In Danish we may have noun phrases like

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

I will not get into more detail about Danish grammar).

The strict rules, like killer rules, can delete structures 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 the corresponding rule could look like

```
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 look

```
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).

This will become a problem when we get to bigger systems and more complicated structures, cf. below.

Basic ideas about *generators and translators:*

Generators are context free rules with annotations, as described above.
Translators are
1) one-shot and 2) compositional.

The fact 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 wrt. wellformedness).

Compositionality in fact means that the image of a complex unit 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 as the one we just saw, which manipulates 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, change dominance, delete nodes, and insert nodes.

As mentioned above the definition of the role of the translator was not optimal. Therefore in the course of the spring 1987 work on a slightly different version of the same basic 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 just saw, 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 modularisable.

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.

So if

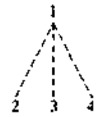

Fig. 1

is the input from the previous level then the shape of the object may e.g. be one of the following after application of the generator

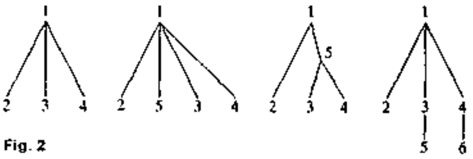

Fig. 2

The software currently used is a prototype which is being developed by a cooperation of computer scientists at various EUROTRA centres. It runs under UNIX and the programming language is basically Prolog. The EUROTRA centres are using a wide variety of computers.

The definition of the levels

The Interface Structure is the common exchange format between the various languages. This means that for all languages the same legislation has to be used, and this legislation has to cover all the common phenomena.

For the other levels of representation ERS, ECS, EBL as well we have common definitions. But these are of a different nature: the closer we get to surface level, the more the languages differ and therefore the definitions of the lower levels are less detailed and have to be followed less rigorously than the definitions of the higher levels.

I will now briefly show the steps of a translation of a simple sentence from Danish ECS to German IS

As can be seen (Figure 3) the ECS structure reflects the surface word order. The Danish input sentence is "I 1982 blev alle forslagene vedtaget af Kommissionen".

At ECS the constituents are built.

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

Then finally at IS (Figures 5 and 6), the case roles of the various constituents are determined. Surface phenomena like 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 (ARG1, ARG2,...). It is complicated to define case roles (like AGENT, EXPERIENCER, ORIGIN, SOURCE, ...) in a way that counts for all languages, this is the reason that for the time being we are using this very simple set of roles, which is then supplemented by lexical semantic features.

As can be seen from Figure 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 figures).

Comments to the trees above: these trees are the output from the first small implemented system (February 1987). The inconsistencies in naming etc. have been removed since.

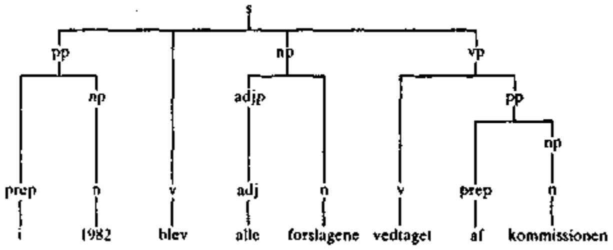

Fig. 3 Danish ECS

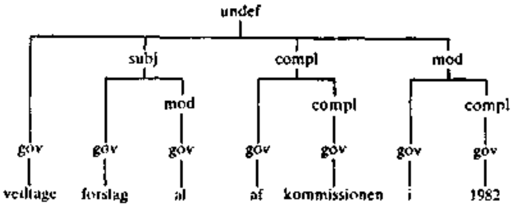

Fig. 4 Danish ERS

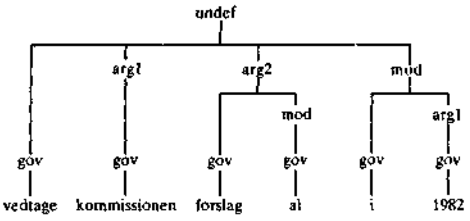

Fig. 5 Danish IS

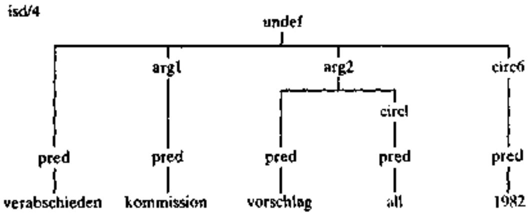

Fig. 6 German IS

Comments on the lexical entries at IS

For the distinction between different readings of ambiguous words lexical features of the semantic kind are needed, i.e. features of the type human-nonhuman, concrete-abstract, and all the subject field features known from ordinary dictionaries, like zoological, medical, etc. Here we shall 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 is word senses that fall within the subject field of IT and neighbouring fields, as well as the general senses. Neighbouring fields are in Community terms: administration, economy, legal aspects... I.e., 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, by the semantic features of (one of) its arguments etc., it is split up. There are words however that do not in a reasonable way lend themselves to a monolingual disambiguation.

In such cases there are 2 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

It should be stressed that the monolingual solution, i.e. disambiguation either in analysis or in generation, is the preferable, because the transfer component should be kept as small and simple as possible.

One of the problems in the lexical transfer is the definition of a lexical unit: 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 diverging: some people would like to do the translation the most elegant way, and that would in our case be to split everything into small units which can be translated by simple transfer and then recombined by the target language grammar in a 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 "lexicalised" and 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 lexicalised.

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

One of the reasons that 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. But this is not the heart of the problem, the problem is the non-compositional or 'context-sensitive' translation and how to handle it.

Take as an example (Danish-French-English)

handelsoverskud	excédent commercial	trade surplus
handelsminister	ministre du commerce	minister of commerce

Here the Danish noun *handel* translates into two different word classes in French and into two different lexical units in English.

The status of the project summer 1987

The Council Decision of 1982 divides the project period into three phases. The first phase is preparatory and has been finished; 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 2,500 lexical entries. The second phase finishes 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, both 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 made for the Spanish and Portuguese languages, as these became part of the project only recently.