Unification Based Transfer

Björn Beskow
Uppsala University

Abstract

This paper concerns formal tools and methods for describing the transfer phase of a machine translation process. The unificational framework of the MulTra formalism is elaborated. The transfer relation representing a relation between translation units is formally defined. A control structure based on specificity is introduced, and shown to provide a way of treating some linguistic phenomena.

1 Introduction

1.1 The MulTra system

The MulTra system is being developed within the project "Multilingual Support for Translators and Writers". The project is sponsored by Digital Equipment Corporation and by NUTEK. It runs in tandem with a sister project at IPPI in Moscow.

The MulTra system is a support system for translating and writing. The system is fully interactive, where the translator or writer always is in command. The informational domain of the system is not just text, but consists of several different kinds of elements, ranging from ordinary text to layout and 3D graphics. The system thus operates in a *Compound Document Environment*.

One functionality of the system is a Machine Translation component. The translator or writer can use this component by selecting an element or a region of the document and querying the system for a translation.

The transfer approach to machine translation has been adopted. Analysis is performed by a chart parser producing feature structures (see [Sågvall-Hein 1987]). Synthesis is performed by a unification-based generator taking feature structures as input. This paper will focus on the transfer phase of the MulTra system.

1.2 Transfer Process

Techniques and tools for analysis within the field of Machine Translation (MT), and during the last years to some extent also generation, are reasonably well understood and developed. The transfer process, on the contrary, is still often described as a black box. Very few descriptions of the transfer component in an MT system has been given. If the transfer process is described at all, the description is almost always procedurally formulated, and sometimes even hardcoded. This leads to several unfortunate consequences. The *translation relations* which the transfer rules are supposed to describe, becomes hidden behind sequentially ordered transformations. The grammar writer is

Proceedings of NODALIDA 1991, pages 31-39

forced to try to foresee complicated interaction between rules. The maintainability and extendibility of the grammar becomes poor.

During the last years, though, some efforts have been made to describe the transfer process within a unificational framework (see for example the ELU project in Geneva Estival *et.al.* 19901 or Karlsson *et.al.* in Helsinki [Carlsson & Vilkuna 1990]). This is also the approach of the MulTra system. For reasons of transparency, a transfer formalism should be declaratively formulated. The formalism should be transparent, in both an external and internal sense. The possible interaction between rules should be fully controlled by the formalism. Rules should be stated in a linguistically intuitive way.

The unificational paradigm has shown to be fruitful within analysis and generation, and has become a de facto standard in the research community. Formalisms within the paradigm is characterized by their declarativeness and transparency. The MulTra formalism is thus formulated in terms of unification and subsumption.

I will first describe the declarative formulation of the MulTra formalism. Discussing the procedural reading of the formalism will lead me to the issue of control. I will show how a control structure based on the notion of specificity can be defined within the unificational paradigm.

2 Translation Relations

The purpose of a transfer component is to capture the translation relations that may hold between units in two languages (see [Wikholm 1991] or [Ingo 1991]). These units can be of different type. A translation relation can hold between two *lexemes,,* like 'rock' in English and 'klippa' in Swedish. It can hold between two *structural(units,,* phrases (see [Wikholm 1989]), like for example a noun phrase consisting of a head and a prepositional phrase in English, and a phrase with the same structure in Swedish. It may also hold between combinations of lexical and structural units, like core phrases (see [Sågvall-Hein *et.al.* 1990] or [Östling 1991]), or cases where a lexeme in one language is related to a phrase in the other language.

These different cases have in MT traditionally been referred to as a distinction between *lexical* and *structural* transfer. But since the different cases are conceptually equal on the level of translation relations, the notion of translation relations implies that no sharp distinction should be made between structural and lexical transfer. You should be able to describe them with the same type of rules.

The basic building blocks in unification-based formalisms are *feature structures* (see [Shieber 1986]). A feature structure is an unorded set of feature value pairs. Feature structures provides a unifolm framework for representing translation units. A translation unit, be it a lexical or a complex unit, is in the MulTra formalism represented by a feature structure. A transfer relation is thus a relation between two feature structures.

3 Transfer Rules

The relations between translation units are defined by transfer rules. These rules describe the feature structures representing the translation unit being related, and places restrictions on these structures.

3.1 Syntax

Feature structures are usually described by a set of identity equations holding over the feature structure being described. An identity equation states what value should be for a specified feature. The special symbol '*' usually refers to the 'root' of the feature structure being described. In the MulTra formalism, an identity equation may contain variables, prefixed by a questionmark '?'.

These are not variables in the feature structure being described, but in the description itself. A variable thus denotes a substructure. A special variable is 'ANY' which can be viewed as the anonymous variable.

A transfer rule consists of four parts:

- A label
- A set of identity equations which describes the source feature structure.
- A similar set of identity equations describing the target feature structure
- A set of transfer equations relating variables mentioned in the source and target identity equations. A transfer relation states that the transfer relation holds between the subparts of the source and target structure denoted by the variables.

Figure 3-1 below is a simple example of a lexical rule. It describes the relation between two structures representing lexemes in Swedish and German. It says that a structure with the value 'montering' on the *lex* feature stands in the translation relation to a structure having the value 'montiren' on the *lex* feature.

It does not contain any transfer relations. It is thus an 'atomic' rule.

```
Label
    Montering
Source
    <* lex> = montering
Target
    <* lex> = montieren
Transfer
    {}
```

Figure 3-1 Lexical rule

Figure 3-2 below is a little more complex rule, describing the translation relation between two prepositional phrases. It relates a structure having the substructure denoted by ?prep1 as value on the *prep* feature, and the substructure denoted by ?rect1 as the value on the *rect* feature, with a structure that has ?prep2 as value on the *prep* feature and ?rect2 as the value of the *rect* feature. It also says that the substructures denoted by ?prep1 and ?prep2 stands in transfer relation to each other, and ditto for ?rect1 and ?rect2.

```
Label
    PP
Source
    <* prep>    = ?prep1
    <* rect>    = ?rect1
Target
    <* prep>    = ?prep2
    <* rect>    = ?rect2
Transfer
    ?prep1      <=> ?prep2
    ?rect1      <=> ?rect2
```

Figure 3-2 Structural rule

Figure 3-3 below is an example of a hybrid rule relating structures with both structural and lexical information. It describes the relation that holds between a structure representing an NP with the lexeme 'Whisky' as its head, modified by the PP 'on the rocks', and a structure representing an NP with the lexeme 'Whisky', modified by the PP 'med is'.

Since the transfer rules relates feature structures, not lexemes or trees, the grammar writer may take an arbitrarily large context into account.

```
Label
      whisky-on-the-rocks
Source
      <* head lex>          = 'whisky'
      <* prep lex>          = 'on'
      <* rect def>          = def
      <* rect det lex>      = 'the'
      <* rect head lex>     = 'rocks'
Target
      <* head lex>          = 'whisky'
      <* prep lex>          = 'med'
      <* rect head lex>     = 'is'
Transfer
      {}
```

Figure 3-3 Hybrid rule

3.2 Semantics

To this syntax, we have to add some semantics to say what it means that two feature structures stands in the transfer relation. Given two feature structures S and T, we start by defining the set of *applicable* rules wrt. S and T. We say that a rule is applicable iff its source part subsumes S. Let A be the set of applicable rules as defined in 3-1 below:

A = { r: source(r) subsumes S }

Definition 3-1 Applicable rules

The use of subsumption. in the definition will guarantee that the transfer process be sound. No feature can be added to S by any rule.

We then define the set of actually *applied* rules to be a subset of the set of applicable rules A. with the additional demands that

- the target part of the rules be unified with T and
- all the transfer equations in the rules holds.

Let a be the set of applied rules as defined in 3-2 below:

a= { r: source(r) subsumes S &
target(r) unifies with T &
$\forall$ e in equations(r): e holds}

Definition 3-2 Applied rules

Note that the definition of applied rules relies on the transfer relation yet to be defined.

Now we shall define the *most general* feature structure that unifies with the source part of all the applied rules. Let C be a structure as defined in 3-3 below:

$\forall r \in \alpha$: C unifies with r.

Definition 3-3 Completeness

C will help us guarantee that the transfer process is complete, by acting as a 'record' of all applied rules.

We are now ready to formally define the transfer relation. We say that the two feature structures S and T stand in the transfer relations, if and only if there exists a C, such that the source structure S subsumes C:

S <=> T if and only if

$\exists$ C as in Definition 3-3 : S subsumes C.

Definition 3-4 Transfer Relation

The definition implies that, if S is not the empty structure, there exists a empty set of applied rules, and T is the unification of all target parts of the applied rules.

This is a recursive definition of the transfer relation, and the recursion is introduced above in the definition of applied rules. It is recursive over sub-parts of the structures S and T, mentioned in the rules. The recursion eventually ends with atomic rules, which is to say, rules that lack further transfer equations.

The transfer process, or the actual computing of a transfer relation, consists in creating a constructive proof that the transfer relation holds. The procedural reading of the transfer process follows readily:

- Given a source structure S and an underspecified target structure T:
- Compute the set A of applicable rules wrt. S.
- Apply the rules in A in parallel, by unifying the target parts with T and recursively compute the transfer equations, yielding the applied rules a.
- Compute the structure C by unifying the source part of all applied rules in a.
- Verify that S subsumes C.

From the definition follows completeness of the transfer relation. Since C is the unification of the source part, and S subsumes C, all features of the source structure have been considered in the transfer process.

It can also be seen from the definition that the transfer process will terminate, since the recursion introduced in the transfer equations relate substructures.

4 Control

The definition of the transfer relation implies that *(any subset* of the applicable rules which yields an appropriate C could be applied. To this general framework, we will add a control structure specifying which applicable rules to apply. We will base this control structure on the notion of *specificity*.

4.1 Specificity

Specificity is a general heuristics well understood and used in artificial intelligence, and elaborated by for example the ELU project in Geneva ([Estival *et.al.* 1990]). The basic idea is that a more specific rule should precede or block the application of a more general rule.

Specificity can be defined in terms of subsumption between the source parts of two rules. A rule A is more specific than a rule B if the source part of B subsumes the source part of A.

If the rules are equal by subsumption, a rule is more specific if it contains more transfer equations.

A rule A is more specific than a rule B if
- $\text{source}(A) \neq \text{source}(B)$ and
  source(B) subsumes source(A), or
- $\text{source}(A) \equiv \text{source}(B)$ and
  $|\,\text{transfer}(A)\,| > |\,\text{transfer}(B)\,|$

Definition 4-1 Specificity

Specificity thus constitutes a partial order on the transfer rules. This partial order is independent of the source and target feature structures S and T.

We then have to modify the definition of the set of applied rules to reflect this principle of specificity. We constrain the set of applied rules further by saying that if a rule r is in the set of applied rules, no other rule s which is more specific than r is in the set. If a rule is applied, we guarantee that no other more specific rule also is applied. So, let α be the set of applied rules as defined in 4-2 below:

```
α = { r:  source(r) subsumes S &
          target(r) unifies with T &
          ∀ e in equations(r):
                     e holds &
                     ¬∃s ∈ α : s is more specific than r}
```

Definition 4-2 Applied rules

This definition can be viewed strict, as actually prohibit the application of less specific rules. It may also be used to define a *preference ordering* on the transfer relations. A preference ordering is thus a partial ordering on the results of the transfer so that a more specific reading is preferred or precedes a more general interpretation.

4.2 Blocking

This control strategy can be used to handle for example blocking phenomena in an elegant way. Blocking occurs in cases where there is a specific translation which one wants to block a more general interpretation. For example, consider the translation relations in figure 4-1 below:

'Whisky on the rocks' <=>

'Whisky med is'

'Whisky' <=> 'Whisky'

'on' <=> 'på'

'rock' <=> 'klippa'

Figure 4-1 Translation relations

'Whisky on the rocks' we prefer to translate into Swedish as 'Whisky med is'. But 'on' in isolation can be translated as 'på', and 'rock' as 'klippa'.

The general interpretation of an English NP consisting of a noun and a PP is described by the rule shown in figure 4-2 below, which just relates the noun and the prepositional phrase by the transfer equations.

```
Label
          General-np
Source
          <* head> = ?head1
          <* prep> = ?prep1
          <* rect>  = ?rect1
Target
          <* head> = ?head2
          <* prep> = ?prep2
          <* rect>  = ?rect2
Transfer
          ?head1     <=> ?head2
          ?prep1     <=> ?prep2
          ?rect1     <=> ?rect2
```

Figure 4-2 General NP rule

The application of this general rule can is blocked by the rule previously described above in figure 3-3, since that rule is more specific than the general rule in figure 4-2. The source part of the rule in figure 3-3 describes the *lex* value of the head and the preposition, and the *rect*. Thus the source part of the general rule subsumes the source part of the rule in figure 3-3.

4.3 Lexicalized Phrases

As mentioned above, since the transfer relation is a relation between feature structures, the grammar writer can take an arbitrarily large context into account when folmulating a transfer rule. The notion of specificity in conjunction with this possibility of choosing context, also provides for a way of treating a similar class of notorious problems in machine translation, lexicalized phrases and idioms. Consider for example the English phrase 'kick the bucket'. It is a lexicalized phrase, which should be translated to 'dö' in Swedish. The main verb of the phrases may be inflected. This translation relation is described by the rule described in figure 4-3 below.

The rule relates an English verbal phrase whose head verb is 'kick' and whose object is 'the bucket', with a Swedish verb phrase whose head verb is 'dö'. The two phrases share value on the *tense* feature. This rule is more specific than a general reading of the verb phrase, and thus blocks the general reading.

```
Label
      kick-the-bucket
Source
      <* cat>           = vp
      <* tense>         = ?tense
      <* head lex>      = 'kick'
      <* obj def>       = def
      <* obj det lex>   = 'the'
      <* obj head lex>  = 'bucket'
Target
      <* cat>           = vp
      <* tense>         = ?tense
      <* head lex>      = 'dö'
Transfer
      {}
```

Figure 4-3 Kick the bucket

5 Conclusions

In this paper, the transfer formalism of the MulTra system has been described. Feature structures have been shown to provide a uniform way of representing translation units. A declarative formulation of the transfer relation have been stated in terms of unification and subsumption. The procedural reading of this formulation have been elaborated.

Further, the notion of specificity have been discussed, and a control structure based on this principle have been introduced. It has been shown how this control structure provides a way of treating some interesting linguistic phenomena. This raises further questions about on which level in the translation process these phenomena should be treated that is, in the analysis or in the transfer phase. Considerations like this remains to be investigated.

It may also turn out that the principle of specificity is a too powerful heuristics. But that would in any way lead to further insight into blocking phenomena, and might also reveal other interesting generalizations to be done in this field.

In conclusion, I have claimed that the unificational paradigm provides a very suitable framewolk for describing the transfer process, and shown how such a description can been made.

6 References

Carlsson, L. & Vilkuna, M. *Independent Transfer using Graph Unification* COLING 1990

Estival, D. *et.al.* A *Syntax and Semantics for Feature-Structure Transfer* in Third Conference on Theoretical and Methodological Issues of MT, 1990

Ingo, R. *Från källspråk till målspråk Introduktion i översättningsvetenskap "From* Source language to Target Language", Lund, 1991

Shieber, S. *An Introduction to Unification-based Approaches to Grammar.* Volume 4 of CSLI Lecture Notes. CSLI, Stanford, 1986

Sågvall-Hein, A. *Parsing by means of Uppsala Chart Processor*, in Bolc, L. (ed), Natural Language Parsing Systems, Berlin, 1987

Sågvall-Hein, A., Cederholm, Y ., van Stam Rydehell, M. & Wikholm, E. *Fraser i kärnlexikonet.* "Phrases in the Core Vocabulary", Uppsala University, 1990

Wikholm, E. *Sammansättning som översättningsenhet,* "Compound as Translation Unit", Gothenburg University, 1989

Wikholm, E. *Übersetzungstheorie und maschinelle Übersetzungen* "Translation Theory and Machine Translation", Scandinavian Conference of Computational Linguistics, Bergen, 1991

Östling, A. *A Swedish Core Vocabulary for Machine Translation* Scandinavian Conference of Computational Linguistics, Bergen, 1991

Björn Beskow
Uppsala University
Department of Linguistics
Box 513
S-751 20 Uppsala
E-mail: beskow@ling.gu.se