A TWO-WAY APPROACH TO STRUCTURAL TRANSFER IN MT

Rebecca Root
Linguistics Research Center
University of Texas
P.O. Box 7247
Austin, Texas 78712
USA

Abstract

The METAL machine translation project incorporates two methods of structural transfer - direct transfer and transfer by grammar. In this paper I discuss the strengths and weaknesses of these two approaches in general and with respect to the METAL project, and argue that, for many applications, a combination of the two is preferable to either alone.

1. Introduction

One of the central design questions in machine translation is that of the best method of structural transfer, that is, the conversion from the syntactic analysis structure of the source language to the syntactic generation structure of the target language. Although several of the various approaches to this - interlingua, transfer grammar, and direct transfer [Slocum, 84] - share a number of properties which render a choice among them of relatively little consequence, there is at least one point of variance that can have significant practical ramifications. This is the choice between the use of an independent grammar, as one finds in the interlingua and transfer grammar approaches, and direct transfer, where transfer specifications are tied directly to source language structures. Since each method has its advantages and disadvantages, there is no basis for favoring one over the other, independent of a particular application. However, it is highly likely that for a system with any significant range of application, neither approach will be completely satisfactory. Furthermore, decisions made in the design of other components of the system may render a homogeneous approach to transfer impractical. For both of these reasons, we have implemented in METAL a scheme for transfer which is sufficiently flexible to allow for the use of both direct transfer and transfer by grammar. This is done in such a way as to put control of the interaction in the hands of the grammar writer, allowing him to take maximum advantage of the strengths of each approach.

In the following, I will contrast the strengths and weaknesses of the two methods mentioned above and illustrate how a combination can inherit the advantages of each by discussing our experiences with a combined system in METAL. For the sake of clarity, I will first give an overview of the METAL architecture.

2. Overview of METAL

METAL is a machine translation system designed for the translation of technical texts. Currently, it is implemented for German to English translation, but preliminary work has begun on other language pairs. These efforts indicate that, by and large, the design is suitable for application to multiple source and target languages, and work is in progress to make this completely so.

Translation proceeds in three phases: analysis, integration, and transfer. The analysis phase consists of parsing the input sentence and building a phrase structure tree annotated with various grammatical features. Anaphoric links are resolved during the integration phase [Weir, 1985]. During the transfer phase, the parse tree is structurally and lexically modified according to target language specifications. The output sentence is gotten by reading the terminal nodes of this tree.

Our basic method of structural transfer is a fairly direct transfer. Rather than using a separate transfer grammar, transfer instructions are associated with each rule of the analysis grammar. When an analysis rule applies to build a node, stored on that node, along with grammatical features, is the set of transfer instructions associated with that rule. After integration, the selected parse tree is traversed from top to bottom, executing the transfer instructions associated with each node. The instructions typically consist of such things as feature passing, constituent reordering instructions, tree traversal messages, and lexical transfer instructions. Since the grammar writer chooses what transfer instructions to include and how to order them, he has significant control over the flow of the transfer procedure. An example of such a rule is given here. This is a rule for parsing German prepositional phrases. I have left out the various TEST, CONSTRuction and INTEGRation instructions relating to analysis and integration. See [Bennett, 1983] for a complete description of the grammar component. Comments explaining the English transfer instructions are given in italics.

```
PP              PREP  NP
                1     2

TEST            .
CONSTR          .
INTEGR          .
ENGLISH
  (SEF 1 CA GC)   father's CAse becomes first
                  son's Grammatical Case

  (XFR)           transfer the sons,
                  i.e. descend the tree

  (AND
   (INT 1 PO POST)  if first son has POsition POST,
   (XFM FLIP))      make it follow the second son
```

The preposition's value for GC is updated because this can resolve English transfer ambiguities. After this modification, the sons are transferred according to the English instructions found on their nodes. After transfer, the preposition, now with English features because the node has been transferred, is checked for its position requirements. If it is a postposition, it is placed after the noun phrase. For example, the structure associated with the phrase "vor einer Woche" is modified to yield a structure reflecting the the phrase "a week ago". When other target languages are included, their transfer instructions will appear in this rule as SPANISH, CHINESE or whatever. In this way, one analysis could simultaneously serve as input to the transfer procedures for several target languages.

The type of direct transfer described here has several good points. It is very efficient because there is no time wasted in trying rules which don't apply. By the same token, it is fairly easy for the linguist to guarantee the results of the transfer process because he can gear his rules to very specific structures. For example, there are several German constructions which are analyzed by rules with a phrase structure specification NP -> NP NP. One of these is the genitive construction, as in "ein Teil des Programms". The English transfer set associated with this particular rule contains instructions to insert the English genitive marker "of" so that the translation becomes "a part of the program". There is no wasted attempt to make this insertion in the similar, but not genitive, constructions. Likewise, transfer procedures peculiar to those structures are not applied in vain to the genitive construction. As one might suppose, this method also has the real, if somewhat embarrassing, advantage of allowing for fairly easy implementation of ad hoc solutions, which, unfortunately must be resorted to from time to time.

There are, of course, several disadvantages to doing things this way. If there are multiple source languages, the linguist must repeat, in perhaps non-trivial ways, the same target language information for each source grammar. There is no convenient way to state more global linguistic facts that don't relate to immediate constituent structure (this is a problem for analysis as well). Also, this method forces the description of the target language to be made in terms of the constituent structure of the source language. All of these are problems which are better handled in a grammar based approach to structural transfer. Our decision to incorporate a transfer grammar grew out of the need to overcome the last two restrictions, particularly in the treatment of clauses.

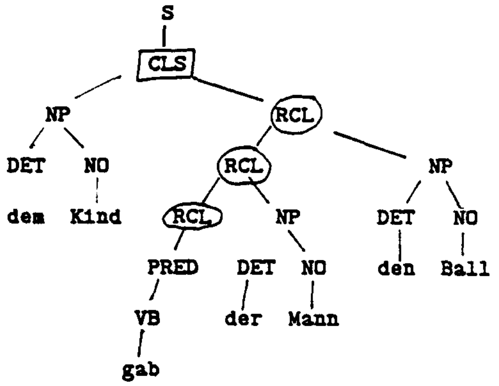
3. The use of transfer grammar in METAL

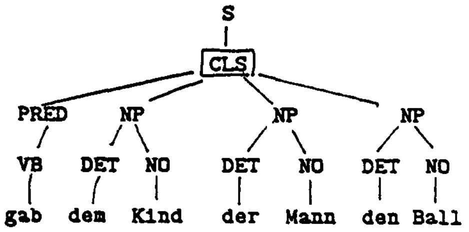
The most pressing need for grammar based transfer was the result of the adoption of a canonical clause structure. The original impetus for using a canonical structure was the need for an efficient analysis of the German clause. However, this canonical structure is put to use by METAL in another way, one which will, in all likelihood, insure its utility, or at least its necessity, for all source languages. The area which would require this is lexical transfer.

Because the dependency between a verb and its object can influence greatly the lexical and structural transfer of both, as well as the structural transfer of the clause as a whole, it is very useful to do a certain amount of lexical transfer, in particular, verb transfer, at the clause level, where both the verb and its arguments are available for inspection and manipulation. This is not a new idea. What is important here is that, although the grammar writer determines when and how clause level lexical transfer takes place, the proper functioning of the transfer procedure depends on the canonical structure of the clause. See [Bear, 1983] for a complete description of the lexical transfer process. The structure we employ is a flat structure, consisting of a PREDicate node followed by one or more arguments:

```
    <clausal category
     /      \     .    .    .
    /        \     .    .    .
  PRED      ARG1   (...) (...)  ARGn
```

However useful a canonical structure is for analysis and lexical transfer, and, in principle, for structural transfer, it creates problems for our direct, node by node structural transfer. The effect of transforming during analysis and integration is that the constituent structure that is reflected by the analysis rule is by no means the constituent structure that actually exists at transfer time for the node built by that rule. This can be illustrated by the following two trees for the sentence "dem Kind gab der Mann den Ball". The first is the parse tree that would have been built if the tree had not been transformed. The second is the actual tree that is built. The circled nodes are ones which are eliminated by flattening, the boxed node is one whose sons have been changed.





Obviously, the transfer portion written for the rule giving the boxed node, CLS -> NP RCL, can have very little specific to say about the transfer process because the actual sons and their order are not at all predictable from anything in this rule. The power to make the various examinations and permutations necessary to execute an appropriate transfer does exist, but they can only awkwardly be specified. Furthermore, they would necessarily be repeated throughout the grammar. The flattening described here takes place in the construction of all clause type structures, and so this same crop of sons could be found hanging on a wide variety of trees. Rather than forcing such a treatment, we exploit what is known about the canonical structure to reap the benefits of treating what is essentially an interlingua as such, by manipulating its structure through the application of transfer grammar rules. This is done in the following way.

Transfer rules are implemented as packages of instructions, typically including tree transformations, of the type found in the target language portion of an analysis rule. However, rather than being stored on a node by virtue of that node's parse history, they comprise an independent portion of the system and are invoked by instructions in target packages. Transfer rules are stored according to one or more root categories. Rules pertaining to a particular category are invoked when the target package associated with a node of that category invokes ORO, the program which accesses the transfer grammar. Because this program is called directly from the grammar and under control of the grammar writer, the overall transfer efficiency is not degraded by the use of a transfer grammar. Any additional cost associated with the use of this grammar is born locally by the constructions which directly benefit. The transfer package associated with the boxed node is given here:

```
CLS          NP  RCL

ENGLISH
 (CLSXFR) do main verb transfer
 (ORO)    invoke grammar rules
              for this category
 (XFR)    descend the tree and transfer sons
----------
```

An example of one transfer rule which ORO would invoke is given below. The first line is a list of root categories to which this rule applies. Thes rule tests to see whether the clause is indicative, and if it is, invokes a transformation by means of the function XFM to place the subject NP before the main verb. The structural description of this transformation is met if the first son is of category PRED and if there is some son following it of category NP and having the value SUBJ for the feature ROL, i.e., some noun phrase fullfills the grammatical role subject. The description allows for the possibility of zero or more constituents preceeding and/or following the NP.

```
CLS CLS-SUB LCL RCL CLS-REL

(AND
 (INT 1 MD IND SUB)
                    if PREDicate is INDicative
                    or SUBjunctive MooD,

 (XFM       ..move SUBJect in front of PREDicate

  (&:1 (PRED:2 -:3 (NP:4 NIL (REQ ROL SUBJ)) -:5))
  (&:1 (NP:4 PRED:2 -:3 -:5)))))
----------
```

There are a variety of rules for placement of other clause constituents. The results of the call to ORO at the clause level is then a tree whose major constituents reflect English word order. Transfer of the constituents themselves is then accomplished by descending the tree in the usual manner.

The discussion above involves only changes which reorder constituents. The transfer grammar also includes rules for more drastic structural changes, such as placement of the particle "not" and the subject of questions within the English verb auxilliary.

4. Summary

We have, so far, only utilized the transfer grammar in places where a direct approach would lead to extreme redundancy in transfer with respect to one language pair. Our treatment of English clauses, however, also has the advantage of reducing redundancy across source languages, since the requirements of the transfer lexicon insure that the input structure to these rules would remain the same. It is likely that further work in other language pairs will give rise to other uses of the transfer grammar.

It might well be asked whether there will be any role for direct transfer in a multilingual system, if it has been found to not be completely satisfactory in a bilingual one. I tend to think there will be, although the role will, no doubt, be reduced. There will probably always be the need for ad hoc solutions to isolated transfer problems, and there is no reason why such non-general solutions should not take advantage of the efficiency available by a more specific direct transfer. And at the very least, this method offers an excellent way to give the linguist control over the flow of the transfer process. The combined capability is particularly valuable when one considers not only the requirements of a completed system, but those of a system still under development, as well.

REFERENCES

Bear, John. "Aspects of the Transfer Component of the METAL Machine Translation System," unpublished manuscript, 1984.

Bennett, Winfield S. "The LRC Machine Translation System: an Overview of the Linguistic Component of METAL," Computers and Artificial Intelligence, vol. 2, no. 2, April 1983.

Slocum, Jonathan. "Machine Translation: It's History, Current Status and Future Prospects", COLING, 1984.

Weir, Carl. "Anaphora Resolution in the METAL Machine Translation System," unpublished manuscript, 1985.