A MODEL FOR TRANSFER CONTROL IN THE METAL MT-SYSTEM

Juan A. ALONSO

Siemens CDS
c/Luis Muntadas, 5
CORNELLA, 08940-BARCELONA, Spain

Abstract

The present paper tries to outline a model to enhance the transfer control within the METAL Machine Translation System. The model is being currently tested in the German-Spanish system which is under development in Barcelona and relies upon techniques belonging to the GPSG framework. The central idea is to extract from the transfer part of the phrase structure rules currently used by METAL all the relevant generalizable information about feature traffic and control dependences, and put it in form of language-dependent tables. This information is then accessed and handled by a few high-level rule operators, called during the transfer process, implementing three general feature principles. The grammar writer is thereby relieved from the tedious task of controlling all the feature traffic between nodes, this resulting in a clearer, shorter and safer grammar for the system.

1.- Introduction.

This paper presents a proposal for the application of some GPSG-based techniques on the METAL MT system with the aim of endowing the system with a stronger control in the Transfer Phase.

The central idea upon which this Transfer Control model is based is to provide the grammar writer with a confortable and safe means for keeping as much feature traffic as possible controlled in the Transfer Phase of the translation process in METAL. Currently, any kind of feature traffic between nodes must be explicitely stated in the rules, and the same happens with the child node transfer process (Control Dependences).

It seems reasonable to think that a great deal of both this feature traffic between nodes and the control dependences within the transfer phase could be generalized and stated outside the rules, in form of language-dependent tables which would then be accessed by a few general operators called in the rules (implementing the principles proposed below). The grammar writer would thus be relieved from this task, all this resulting in shorter, clearer and less error-prone rules.

Most of the ideas conforming the model presented here have been taken from the GPSG framework [Gazdar 85]. The original idea was to directly apply the GPSG principles to the system, since, basically, METAL disposes of the necessary structure for it (i.e., it is a PS-based system, and thus, it works with structural descriptions [trees] consisting of bundles of feature-value pairs [nodes]). However, the fact that the GPSG model was originally conceived for analysis (transfer being quite a different problem) and that METAL lacks mechanisms which are central to the GPSG model, like LP/ID rules, metarules, FSD and FCR, etc., showed the unpracticability of such a direct approach. This is why the idea became to adapt some of the ideas offered by the GPSG (mostly the CAP and the HFC universal feature instantiation principles) and re-formulate them so that they can be used for our purposes.

2.- Fundamentals of the METAL system.

METAL is a Chart-Parser-driven Phrase Structure based MT system, which reflects the classic MT scheme of Analysis, Transfer and Generation phases. During the Analysis Phase, METAL builds from each input source language sentence one or more structural descriptions (henceforth trees), consisting of nodes, which in turn consist of a number of Feature-Value pairs (henceforth f-v-pairs). In the Transfer Phase, the trees obtained in the analysis are converted into equivalent trees adapted to the target language needs. After this, the Generation Phase generates the output sentence/s in the target language, using the transfer trees as input.

Apart from the lexical DBs METAL has some 500 PS rules, whose form can roughly be described as follows:

```
<RULE-IDENTIFIER>
NODE-STRUCTURE  <e.g. "NP --> DET NO">
TEST      <Tests on nodes to be satisfied in order for the
           rule to apply>
          ...
CONSTR    <Analysis-Tree Construction part>   ANALYSIS PART ↑
          ...                                 ----------------
INTEGR    <Anaphora Resolution part>          TRANSFER PART ↓
          ...
TRANSFER  <Transfer-Tree construction part>
```

During the Analysis phase, only the analysis part of the succeeding rules apply, building the analysis tree in a bottom-up manner until a S node is reached. Once the analysis tree has been thus built, the Transfer Phase starts; the transfer part of the rules applied during analysis activates now, climbing down the tree from top to bottom until the terminal nodes are transferred. When the Transfer Process applies on a node N, with f-v-pairs F-V, the child nodes of N are also transferred (and the child nodes of these, and so on, until terminal nodes are reached). Once all the branching nodes dominated by N have been transferred, the transfer process returns control to the father of N, which now bears target-language updated f-v-pairs F-V'.

Two tasks central to the Transfer Process are the Feature Traffic (i.e., which f-v-pairs need to be sent up and down in which moment, and from which node to which node/s), and the handling of Control Dependences (i.e., which child node of one analysis tree or sub-tree [henceforth local tree] must be transferred first, in order for other sibling nodes to be able to be rightly transferred).

The two mentioned tasks are now handled by the Grammar writer in the Transfer part of the METAL Grammar Rules through calls to "low level" feature traffic operators (i.e., copy one or more f-v-pairs from the root node to a child node, from one child node to another sibling node, or from one child node to the root node).

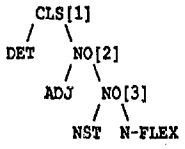
What the present Transfer Control Model proposes is to extract from the rules all which can be generalizable in this process regarding feature traffic and control dependences and carry it out through calls to a few "high level" operators which use information stored in the system database in form of tables stating which f-v-pairs must be present in a given node, which nodes are controllers, and which other nodes are controllees within a given local tree.

3.- Basic Definitions.

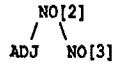
In this section there follows a number of definitions which will be used throughout the rest of the paper.

3.1.- Local tree.

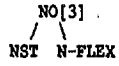
A local tree is a part of a structural description which is currently being dealt with by the particular rule which is under consideration. For example, given an analysis tree of the form



in which the number between [] indicates (for reference) the rule which has built this particular node, the local tree when rule [2] is applied would be:



whereas the local tree for rule [3] would be



3.2.- Types of nodes

Root Node [RN]:

* The root [parent] node of a local tree.

Head Node [HN]:

* In case Control Dependences (see below) exist within the current local tree, the Head Node is the controller node. Otherwise, the Head Node must be explicitely stated for each particular tree structure, normally being the X(BAR-1) child node, in a local tree dominated by a root node X(BAR)

Notice that this definition of Head Node has been tailored ad-hoc for this model and deviates considerably from the traditional notion of Head in the X-bar theory, for instance.

Dependent Node [DN]:

* A child node of the local tree which is controlled by a HN.

Free Node [FN]:

* A child node of the local tree which is not controlled by any HN.

Lexical Node [LN]:

* A terminal lexical node.

3.3- Types of Features

Head Features [HF]:

* A set of f-v-pairs which must be present with the same values both in the Head Node and in the Root Node in some precise moments during the Translation Process.

Lexical Access Features [LAF]:

* A set of f-v-pairs which must be present in the Lexical Nodes nodes prior to their transference into the target language.

Control Features [CF]:

* A set of f-v-pairs which must be copied from the Head Nodes into the Dependent Node/s after the Head Node has been transferred and before the Dependent Node/s is/are transferred.

3.4.-Control Dependence Between Nodes

* A Head Node [HN] controls one or more Dependent Node/s [DN] within a local tree if in order for the DN/s to be properly transferred, it/they must have a set of f-v-pairs whose particular values are to be updated with those values borne by the HN after this node has been transferred.

Basically, our notion of Control coincides with the one given by the GKPS [Gazdar 85]. Control is a language-dependent relationship between nodes, in which there is a controller node and one or more controllee node/s, which ultimately subsumes the concept of agreement (subject-predicate, noun-adjective, etc.). In the Transfer Process, nodes which are controllers must be transferred prior to their controllees, in order to ensure the right agreement between them in the target language.

Notice that a local tree may present different types of Control Dependences (see Fig. 1), with one Dependent Node (the NP[2] node below, dominating a local tree where the NO node controls the DET node for Gender and Number), with two Dependent Nodes (the CLS node dominating a local tree where the NP[$SUBJ] controls both the PRED node for Person and Number and the ADJ node for Gender and Number), or no Dependent Nodes at all (the PP node dominating a local tree where neither the NP node nor the PREP nodes control each other):

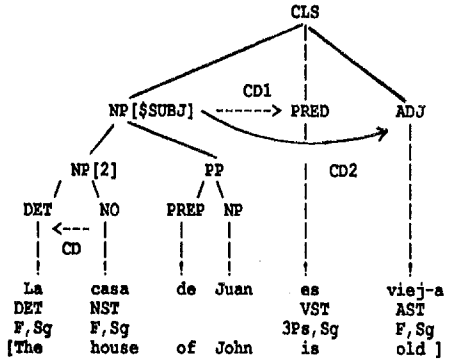


Fig.1: Control Dependences

3.5.- Local Transfer Process

Given a local tree, consisting of a Root Node a one or more child nodes (including a Head Node, and possibly one or more Dependent Nodes, and one or more Free Nodes), we can split the local Transfer Process sequence of the Root Node dominating the local tree into three steps:

* Transfer the Head Node.
* Copying the Control Features (CFs) set from the already transferred Head Node into the Dependent Node/s (if any).
* Transfer the Dependent Node/s, and the Free Node/s (if any).

4.- General Information to be Supplied to the System DB

Basically, three types of information must be stored into the system DB and used later on by the Transfer Process. Physically, this information is implemented in form of a LISP list, although this is purely a parochial programming decision.

4.1.- HEAD Feature List [HFL]:

It contains information stating which f-v-pairs are considered to be members of the set HEAD. The decision of which f-v-pairs must be HEAD members is crucial to the model. In a first approach, we will adopt a pragmatical criterion. This means that we will include as HEAD features those f-v-pairs which are currently percolated in the corresponding rules, and which we make sure are actually needed for the Transfer Process.

However, the aim is to extrapolate from this first approach a (maybe language-dependant) theoretical hypothesis about HEAD Features which enables to state a general criterion to establish the HF membership.

4.2.- Lexical Access Table [LAT]:

It contains LANGUAGE-DEPENDENT information stating which sets of f-v-pairs (Lexical Access Features [LAF]) are needed in order for each (major) Lexical Node to be rightly transferred. In METAL, lexical nodes are transferred by calling the XLX operator in the TRANSFER part of the corresponding rules. XLX takes arguments specifying the needed target language stem retrieval information for a given category, whether an inflexion must be attached to the stem, and the retrieval information for this inflexion. Thus, the LAT table contains information about Lexical Access Features, Inflexion Attachement information (when needed) and Inflection Lexical Access Features (ILAF). In fact, this would be quite similar to the XLX table proposed by Tommy Loomis in [Loomis 87].

Here is a schematic example of the Lexical Access Table:

SPANISH-LAT:

LN	LAF	INF	ILAF
AST	-	A-FLEX	GD NU
DET	GD NU VON	-	-
NST	-	N-FLEX	GD NU CL
PRN	CA GD NU	-	-
VST	MD NU PF PS TN	V-FLEX	MD NU PF PS TN CL
...	...	...	...

AST = Adjectival Stem
DET = Determiner
NST = Noun Stem
PRN = Pronoun
VST = Verb Stem
N-FLEX = Nominal Inflexion
V-FLEX = Verbal Inflexion
A-FLEX = Adjectival Inflexion

CA = Case CL = Inflexion Class GD = Gender MD = Mode
PF = Predicate Form PS = Person TN = Tense
NU = Number

In the case of VST, for instance, the LAT would indicate that, for Spanish, a Verb Stem (VST) Lexical Node must be accessed in the target monolingual lexicon database through the current values of the MD, NU, PF, PS and TN features as keys, that a V-FLEX inflexion must be attached to it, and that this inflexion should be accessed through the current values of CL, MD, NU, PF, PS, TN.

4.3.- Control Type Table [CTT]:

This table contains LANGUAGE-DEPENDENT information stating the Control Dependences for different local trees dominated by different Root Nodes. In this table, for each possible Root Node category (RN), its corresponding Head Node (HN), Dependent Node/s (DN) and Control Features (CF), if any, are specified.

Here is one example of three CTT entries, one for CLS- (with two potential Control Dependences), one for NP- (one CD) and one for PP- (no CDs) dominated local trees :

SPANISH-CTT:

RN	HN	DN	CF
CLS	NP[$SUBJ]	PRED (ADJ)	NU PS GD NU
NP	NO	DET	GD NU
PP	NP	-	-

CLS = Clause
PRED = Predicate
DET = Determiner
ADJ = Adjective
NU = Number
NP = Noun Phrase (BAR 2)
$SUBJ = SUBJECT value of feature ROL
NO = Noun Phrase (BAR 1)
GD = Gender
PS = Person

The first entry specifies that, given a local tree with a CLS root node, its child Head Node (HN) is the NP child node bearing the $SUBJ value for the feature ROL, whereas the PRED child node always is a dependent node. The control features (CF) relevant for this structure are NU (number) and PS (person). Moreover, a sibling ADJ node may also be a Dependent Node, with control features GD (Gender) and NU (Number). This CTT entry controls the subject-predicate, and the subject-predicative_adjective (in copulative sentences) agreement requirements, respectively.

The same should be stated for each grammatical category which may be a Root Node of a local tree (CLS, PRED, NO, etc.).

Two things must be stressed about the CTT table:

* Information about particular feature values can be given to distinguish between categories with the same name (for instance, to ensure that for the subject-predicate agreement, the controller is the NP which bears the role of SUBJect, and not some other sibling NP).
* Each CTT entry must have at least a Head Node specified for each Root Node, and possibly one or more Dependent Nodes, which may or may not be obligatory, with their corresponding control features. This accounts for the possibility of having different local trees dominated by the same Root Node category (the case of CLS, above).

5.- Basic Principles

In a first approach, three Working Principles can be stated for the METAL Transfer Phase. These three principles are actually reflected in the form of three operators (implemented as LISP functions) to be called in the transfer part of the rules.

5.1.- Head Feature Update [HFU]

* The Head Node [HN] of a local tree gets its Head Features [HF] instantiated to the values currently present in the Root Node.

HFU uses the HFL and the CTT table information previously stored into the system DB, and should be applied prior to the local Transfer Process.

5.2.- Control Feature Update [CFU]

* Once the Head Node of the local tree has been transferred, the corresponding Control Features must be copied into the sibling Dependent Node/s before this/these are in turn transferred.

The CFU makes use of information stored into the CTT and the LMT tables, and should be called after the HFU has been applied.

5.3.- Root Feature Update [RFU]

* The Root Node of a local tree gets its Head Features [HF] instantiated to the values of the HF present in the child nodes after these have already been transferred. If any f-v-pair conflict arises (i.e., if for a given f-v-pair two child sons have different incompatible values) the Head Node value will be preferred.

RFU uses the HFL table information, and should be applied after all the local tree child nodes have been transferred.

6.- Transformations

Before and after what we have called the Transfer Process throughout this paper, there may be present one or more transformations which alter the original structure of the local tree to yield the correct constituent structure for the current target language (for instance, most adjectives preceding nouns in German or English must follow them in Spanish)

Two kinds of such transformations can be distinguished, namely "pre-TP transformations" and "post-TP transformations".

Some pre-TP transformations may dramatically change the local tree structure in a way which is not obvious at first sight (deleting, inserting or changing the order of child nodes). Both the CTT table and the CFU function must hold into account this fact and handle automatically every possible local tree structure for each root node category, so that the grammar writer can be thus relieved from another difficult task which very often gives rise to errors in the grammar, namely, the handling in the rules either of new "invisible" child nodes or of old ones with a different order within the tree.

7.- An Example for METAL

Let us see a simplified example of what a typical METAL PS-rule TRANSFER part would look like if the operators implementing these principles were applied. Please, bear in mind that this is a simplified example, and thus, it does not take into account any extra feature traffic which might be present and cannot be generalized by the present model.

Example Rule

```
NP      DET     NO
0       1       2
...     ...     ...
TEST
        ....
CONSTR
        ....
INTEGR
        ....
SPANISH
        (PRE-XFM) ; Tree transformations previous to the TP.

        (HFU) ; Copies the HFs present in NP to the HN node,
              ; which, for an NP Root Node, is the NO node,
              ; as the CTT states.

        (CFU) ; Transfers the NO node (HN).
              ; Copies the Control Features (GD, NU) from

              ; NO to DET, which is the Dependent Node.
              ; Transfers the DET node (DN)

        (XFR n); Transfers Free Nodes if any.

        (RFU) ; Copies all the HFs present in DET & NO to NP

        (POST-XFM) ; Tree transformations after the TP.
```

8.- Conclusion

Although the model outline here must still be fully tested and parts of it re-specified according to the results, it seems to be a valid approach to the problem of the transfer control in the METAL system.

A number of questions still remain open, namely:

* How to deal with trees where Control Interdependences exist. This is the case of German NPs bearing different adjectival inflexions depending both upon the gender of the noun and upon the type of determiner (weak/strong adjective declension)

* How to deal with local trees with more than one Head Node (coordinate structures, for instance).

* Whether some type of GPSG FCR- or/and FSD-like mechanisms (see GKPS [Gazdar 85]) could be used in this model.

* Whether this model is generalizable to other MT systems.

The outlined model may be a starting point to begin introducing some of the techniques offered by the current linguistic theories (GPSG, LFG, GB, etc.) into the MT field, and at the same time trying to bring these pure theoretical models into the practical fields of MT systems already under development.

9.- References

[1].- GAZDAR, KLEIN, PULLUM & SAG. Generalized Phrase Structure Grammar, 1985.

[2].- LOOMIS, Thomas M., Morphological Generation Within the METAL Machine Translation System. Linguistic Research Center, Austin, U.S.A., 1987

[3].- P. ISABELLE, E. MACKLOVITCH, Transfer and MT Modularity. Coling '86 Proceedings

[4].- S. M. SHIEBER, A Simple Reconstruction of GPSG, Coling '86 Proceedings.