

Interaction between Structural Changes in Machine Translation

Satoshi KINOSHITA
Research and Development Center
Toshiba Corporation
Komukai-Toshiba-cho 1, Saiwai-ku,
Kawasaki, 210 Japan

John PHILLIPS Jun-ichi TSUJII
Centre for Computational Linguistics
UMIST
P.O.Box 88, Manchester,
M60 1QD, U.K.

Abstract

This paper discusses complex structural changes during transfer within a non-destructive transfer framework. Though the description of each individual structural change is not difficult, special provision must be made when they are combined, because interaction between them sometimes causes unexpected problems. Transfer of coordinate structures is also discussed as this sometimes necessitates a structural change and interacts with other structural changes in a problematic way.

1 Introduction

Several focal issues have emerged in machine translation (MT) research as the result of recent intensive studies in the field. That is,

- Declarative Frameworks for expressing Bi-lingual Knowledge
- Integration of Knowledge-based Processing and Contextual Processing with the Translation process
- Effective Exploitation of Domain/Text Type-Specificity (or Sublanguageness) in MT and Discovery Processes for such Specificities

Though new frameworks of MT such as Statistics-based MT (SBMT), Example-based MT (EBMT), Analogy-based MT (ABMT), Knowledge-based MT (KBMT) etc. look radically different from conventional linguistics-based MT (LBMT) such as Transfer-based MT, they address one or two of the above focal issues and ignore the rest[7][10]. In particular, the new paradigms of MT tend to ignore the first issue i.e. declarative representation of bi-lingual knowledge and the complexities involved in the process of combining units of bi-lingual knowledge.

It is our contention that any MT system, whichever paradigm it belongs to, has to store a set of translation-equivalent units for a pair of languages and combine these to produce larger units of translation. In EBMT, for example, a set of translation examples has to be stored and several of them have

to be combined properly to produce translation. Because of the declarative nature of translation examples, EBMT inevitably encounters the same complexities of combining translation units in declarative representation as LBMT does.

Research in LBMT[1][2][3][11][12][13] has revealed that difficulties in the declarative representation of bi-lingual knowledge stem mainly from the treatment of idiosyncratic structural changes caused by lexical items, and interactions of such idiosyncratic structural changes when they co-exist in single sentences. These type of structural changes also cause problems when they are combined with general or ordinary linguistic phenomena such as coordination. A formal framework to cope with these matters is essential in other approaches, such as EBMT, as in LBMT, if the translation is produced in a compositional way.

In this paper, we discuss problems caused by such interaction, and give solutions in our logic-based transfer framework[8]. Our transfer model views the transfer process as deduction; it produces the target linguistic description without changing the source description. This gives a clear contrast to the conventional tree-transducer model, which gradually transforms the source description, and is therefore hard to control.

Because of the logic oriented nature of our framework, it can also be extended easily to cope with the other focal issue of MT such as the integration of knowledge-based processing with translation[4].

2 Formalism

We use a simple formalism, representing a sentence as a network of semantic dependencies. The basic units of the formalism are *indices*, *properties*, and *relations*. A logical form consists of an unordered set of *terms*; each term is either a property predicated of an index, or a relation between two indices. The written notation depicts properties and relations as unary and binary functors, respectively, and indices as their arguments, i.e. within brackets. A logical form representing 'John saw Mary' might be

$e : \text{john}(j) \ \& \ \text{see}(e) \ \& \ \text{mary}(m) \ \& \ \text{tense}(e,\text{past}) \ \& \ \text{subj}(e,j) \ \& \ \text{obj}(e,m)$

This representation is simplified particularly in that

the man who Mary saw: $x : \text{man}(x) \ \& \ \text{mary}(m) \ \& \ \text{see}(e) \ \& \ \text{tense}(e,\text{past}) \ \& \ \text{subj}(e,m) \ \& \ \text{obj}(e,x)$

the man who saw Mary: $x : \text{man}(x) \ \& \ \text{mary}(m) \ \& \ \text{see}(e) \ \& \ \text{tense}(e,\text{past}) \ \& \ \text{subj}(e,x) \ \& \ \text{obj}(e,m)$

I managed to paint my house quickly: $m : \text{me}(me) \ \& \ \text{manage}(m) \ \& \ \text{tense}(m,\text{past}) \ \& \ \text{subj}(m,me) \ \& \ \text{obj}(m,p) \ \& \ \text{paint}(p) \ \& \ \text{obj}(p,h) \ \& \ \text{of}(h,me) \ \& \ \text{house}(h) \ \& \ \text{quick}(q) \ \& \ \text{subj}(q,p)$

Mary looked at the brightly painted house: $l : \text{mary}(m) \ \& \ \text{look}(l) \ \& \ \text{tense}(l,\text{past}) \ \& \ \text{subj}(l,m) \ \& \ \text{at}(l,h) \ \& \ \text{bright}(b) \ \& \ \text{subj}(b,p) \ \& \ \text{paint}(p) \ \& \ \text{obj}(p,h) \ \& \ \text{house}(h)$

Mary's house was painted: $p : \text{mary}(m) \ \& \ \text{of}(h,m) \ \& \ \text{house}(h) \ \& \ \text{paint}(p) \ \& \ \text{tense}(p,\text{past}) \ \& \ \text{obj}(p,h)$

Figure 1: Examples of Logical Forms

the relation *tense* here stands for what should be a complex representation of tense, aspect, and aktionsart, related anaphorically to the preceding discourse. It can be seen that the representation includes the names of the concepts, both objects and events, which are mentioned in the sentence, and the semantic relationships between them. Each of the three indices, *j*, *m* and *e*, is a unique label given to a node in the dependency network. The indices also serve as discourse referents when the logical form is incorporated into a discourse representation. The root node of the network is specified at the beginning of the representation, in this case *e*, the node representing the seeing event. In terms of discourse representation theory, *e* is the discourse referent of which the logical form is a description. As long as the root node of the network is specified, it is possible to convert a logical form mechanically into a structured representation such as a dependency graph or a feature structure such as

predicate : see
tense : past
subject : [predicate : john]
object : [predicate : mary]

Some other examples of logical forms are shown in figure 1. The particular characteristics of the formalism — the types of relations and predicates used and the manner of their use — are justified elsewhere[9]. We only state here that the formalism can treat the phenomena which are treated by traditional formalisms, along with additional phenomena relating to discourse structure and lexical semantics.

3 Transfer

We follow the standard division of a machine translation system into stages of analysis, transfer, and generation. The parts of the system include algorithms for analysis, transfer, and generation, descriptions of the structure of individual languages (used by the analysis and generation algorithms), and descriptions of equivalence relationships between pairs of languages (used by the transfer algorithm). It is a requirement for the transfer description that it should state all and only the equivalence relationships between expressions of a pair of languages. It

should contain no general linguistic information, either universal or specific to any single language: these types of information belong in the formalism and content, respectively, of the descriptions of individual languages. In fact most of our transfer descriptions consist of direct equivalences between predicates or groups of predicates of the language pair.

A transfer rule shows a particular translation equivalence between two languages. It consists of three parts: a pair of logical forms, one for each language, representing the equivalence, and a third logical form giving the conditions under which the equivalence is applicable. We call these the equivalence pair and the condition respectively. Two logical forms form an equivalence pair if the natural language expressions they represent have overlapping denotations (using 'denotation' in a very broad sense to encompass not only the referential possibilities of nouns but the possibility in general of applying a predicate to a particular index). The rule can therefore be used for translating in either direction: the two logical forms of the equivalence pair are always translationally equivalent if the condition is satisfied. The logical forms of the equivalence pair will be indistinguishable from logical forms of the respective languages, using the same predicates and relations. The logical forms of the condition use meta-predicates which allow reference to the logical form of the current local translation unit (sentence or clause) and to the linguistic context. In practice, most transfer rules have no condition; they consist just of an equivalence pair. Some examples of rules follow:

English	Japanese	Explanation
table(X)	teeburu(X)	table = teeburu
paper(P)	ronbun(P)	paper = ronbun
paper(P)	sinbun(P)	paper = sinbun
subj(E,X)	ga(E,X)	subject marker
obj(E,X)	wo(E,X)	object marker
hypothetical(E),like(E)	-tai(F),omou(E)	would like =
iff inLeft(obj(E,F))		-tai omou

The following is the basic definition of transfer between two texts (sentences, clauses, or whatever):

A source logical form and a target logical form are correctly transferred if the terms of each can be divided into non-overlapping subsets such that the source subsets can be placed into one-to-one correspondence with the target subsets by each corresponding pair being 'unified' with the two halves of the equivalence pair

of a transfer rule, and if the accumulated conditions of the transfer rules used are satisfied.

'Unification' in this definition is rather similar to graph unification.

Transfer rules specifying larger sets of terms can be used to state translational equivalences for idioms and fixed expressions. Simple idioms present no particular problem. One translation of the English idiom to *kick the bucket* into Japanese is *sinu* 'die', and this can be represented by the transfer rule

kick(K) & obj(K,B) & the-bucket(B) \longleftrightarrow sinu(K)

Fixed expressions, appropriate only to a specific context, may require conditions which refer to the discourse representation.

4 'Complex' transfer

4.1 Types of complex transfer

Lindop & Tsujii (1991) list and discuss a variety of examples that are always problematic for machine translation systems. We suggest that what makes these examples difficult is that different dependency structures are associated with otherwise synonymous lexical items. We break the problem down into four subparts.

i) Argument-switching as in the translation of the German

Der Wagen gefällt mir — I like the car
The car pleases me

gefallen(E) & nom(O) & dat(S)
 \longleftrightarrow like(E) & subj(S) & obj(O)

In argument-switching, the relations between the lexical item and its dependents are not translated standardly. Although the German nominative normally corresponds to an English subject, it must in this example be translated as a dative.

ii) Head-switching as in the translation of German

Hans schwimmt gern — John likes swimming
John swims gladly

like(L) & obj(L,S)
 \longleftrightarrow gern(L) & subj(L,S) iff shift(L,S)

The German sentence is a description of a situation to do with swimming; the English is a description of a situation to do with liking. The *shift* predicate is explained below.

iii) Decomposition as in the translation of the Japanese

John-ga jisatusita — John committed suicide

jisatu_suru(E)
 \longleftrightarrow commit(E) & suicide(S) & obj(E,S)

Here the single Japanese word *jisatusuru* is translated as the English phrase *to commit suicide*. Some types of decomposition can give rise to special problems when there are modifiers or dependents involved. These are discussed in detail by Tsujii et al. (1992). Shortage of space and the variety and complexity of examples prevents their discussion here.

iv) Raising as in the translation of the Welsh

Fe ddigwydd fod Siôn yma
happens be John here

John happens to be here

digwydd(E1) & subj(E1,E2) & subj(E2,X)
 \longleftrightarrow happen(E1) & subj(E1,X) & obj(E1,E2)

In the English, the subject of *happen* is John, but in the Welsh, the subject of *digwydd* ('happen') is the situation of John's being here.

The predicate *shift* is introduced to define a translation equivalence where certain dependencies are differently arranged in the target language, as compared to the source language. It can be understood in terms of semantic type-shifting — the types are differently distributed in the source- and target-language logical forms. *Shift*(*H,E*) means that in any instance of *subj*(*H,X*) or *P*(*X,H*) (where *P* stands for any relation) in the logical form representing the English text, the *H* corresponds to an *E* in the logical form representing the other language. The following example shows how the example under (ii) above would work in practice.

Ich weiß Hans schwimmt gern - I know John likes swimming

ich(i)	me(i)	me(i)
wissen(w)	know(w)	know(w)
subj(w,i)	subj(w,i)	subj(w,i)
obj(w,s)	obj(w,s)	obj(w,g)
Hans(j)	John(j)	John(j)
schwimmen(s)	swim(s)	swim(s)
subj(s,j)	subj(s,j)	subj(g,j)
gern(g)	like(g)	like(g)
subj(g,s)	obj(g,s)	obj(g,s)
	{shift(s,g)}	

The columns of the table show the German source logical form, the English logical form as it would be without *shift*, and the actual English target logical form. The horizontal divisions of the table represent the fields of operation of separate transfer rules.

4.2 Interaction between types of complex transfer

When several of these syntactic changes are combined in one example, the logical form resulting from transfer is incoherent. The following examples are types on which most, if not all, existing machine translation

systems will fail, as will the mechanism proposed here so far.

English - Welsh; raising with argument-switching:
John happens to like swimming

fe ddigwydd fod yn dda gan John nofio
happens be nice by John swim
(swimming-being-nice-in-John's-opinion occurs)

Dutch - English; verb - head-switching twice:

Jan zweemt graag toevallig
John swims gladly happeningly

John happens to like swimming
(also 'John likes happening to swim')

There are two separate causes of difficulty in these examples. Firstly, the long-distance movement caused by raising causes problems in the transfer of structures which have been the target of raising. The simplest way to get round the problem is to require syntactically-deleted arguments to be marked explicitly on all the verbs of which they are logical arguments, treating the phenomenon syntactically as an example of lexically-triggered extraction, on a par with the extractions of syntactic objects in a *man I know* and an *easy man to please*. Transfer rules which introduce raising predicates will then have to bring in the new subject explicitly. For instance, the rule for *happen* and *digwydd*, (iv) in §4.1, will be re-written as

digwydd(E1) & subj(E1,E2)
→ happen(E1) & subj(E1,X) & obj(E1,E2)
iff subj(E2,X)

The second point is that the *shift* predicate must be defined in such a way that it can cope with recursive changes in the dependency structure brought about by the occurrence of several interdependent head-switching translations. It seems that *shift* can be made to do this simply by having e.g. *shift(H,E)* affect all instances of *subj(H,X)* or *P(X,H)* (including *shift(X,H)*) not transferred by the transfer rule which introduced the *shift(H,E)*.

Together, these two stipulations enable the transfer of examples involving head-switching, argument-switching, raising, and most types of decomposition.

5 Transfer of Coordinate Structures

5.1 Problems in transfer of coordinate structures

Coordination often requires exceptional treatment in parsing and generation of natural language. Transfer is no exception. Transferring a coordinated structure

sometimes produces a miserable result due to the accompanying combination of structural changes or the fact that coordinated linguistic objects require different target words for a shared source word. However, few attempts have been reported to formalize this problem.

We here divide the problem into two categories:¹

- Divergence in semantic constraints
- Combination of complex transfers

The first type of problem occurs when semantic features of the coordinated source words require a source word to be translated into two different target words. A typical example can be seen in translation of the following English sentence into Japanese.

(1a) She wears a hat and shoes.

(1b) kanojo-ga boushi-wo kaburi, kutsu-wo
she-subj hat-obj wear shoe-obj
haku.
wear

As is understood from its translation, "wear" is translated "kaburu" or "haku" in Japanese, depending on whether its object is something worn on the head or on the foot (or leg). This means that, in this example, coordination of objects in English should be altered to that of verb phrases in Japanese.

This type of knowledge for lexical choice is very common in a transfer or bi-lingual dictionary, and plays an essential role in lexical transfer of most current transfer-based MT systems. The problem is that neither a transfer program or a transfer-rule writer expects such an awkward problem caused by coordination. To translate "wear" into "kaburu" in the above example, a rule writer may usually write the following rule in our notation²:

wear(X) → kaburu(X) iff obj(X,Y)&HAT(Y)

But the condition part of this rule is implicitly expected to be interpreted as follows.

wear(X) → kaburu(X) iff $\forall Y$ obj(X,Y)&HAT(Y)

The precise definition may change depending on how the coordinate structure is represented. But the

¹There is another type of problem which is based on the syntactic differences between coordination constructions in source and target languages. For example, "2 or 3 pounds" in English should be translated "2 pondo ~~ka~~ 3 pondo" (2 pounds or 3 pounds) in Japanese and "dwybunt neu dair" (2 pounds or 3) in Welsh. (The Welsh expression is used only for a sum of money. Another exists for weight.) This type of problem should also be solved in transfer, but we do not mention it here.

²In this section, we consider translation whose source and target logical forms are on the left and right sides of a transfer rule. For the sake of simplicity, transfer rules hereafter are described as uni-directional ones.

point is that “wear” may be translated “kaburu” only if all the objects have a feature of being a “HAT”.

A simple transfer algorithm, for example, may choose the target word when it finds the first applicable transfer rule for a source word: this algorithm may produce “boushi-to kutsu-wo kaburu” for the sentence (1a), which means that the semantic relation between “wear” and “shoes” is ignored. There may be another type of transfer algorithm which simply fails because it cannot provide two different target words to one identical source word.

The second type of the problem occurs when one of the coordinated objects triggers a complex transfer which is described in §3. This type of problem can be seen in the following translation.

He committed murder then suicide.

kare-ga satsujin-wo okashi, jisatsu-shita.
 he-subj murder-obj commit commit-ed suicide

This problem is more complicated than the previous one because complex transfer, in this example “many-to-one transfer”, causes a structural change. Our simple transfer algorithm mentioned in the previous section may produce a disastrous result for this type of translation.

There are several possible solutions to this problem. The simplest one is that a transfer-rule writer writes all the transfer rules which explicitly describe every possible sub-structure with coordination. This is of course unrealistic.

Another solution is to make a transfer program which modifies the transfer result dynamically when, for example, a source word is given two different target words. But such a dynamic modification of the result during transfer is against our policy of logic-based transfer, because this means gradual transformation of the source structure and therefore transfer cannot be formalized as logical inference.

5.2 Transfer with Coordinate Expansion

5.2.1 Coordinate Expansion

Hereafter we concentrate on a case where coordinated objects of a verb cause the problem, though there is apparently an example where other cases such as “subj” cause the same problem.

The basic requirement in logic-based transfer is that coordination of objects should be reorganized to that of verb phrases or sentences, which is not supposed to cause problems in transfer. We call this reorganization “coordinate expansion”.

The following is a part of logical form for (1a), which involves a coordinate structure.

wear(w)&obj(w,o)&coord(o,o1)&hat(o1)&
 coord(o,o2)&shoe(o2)

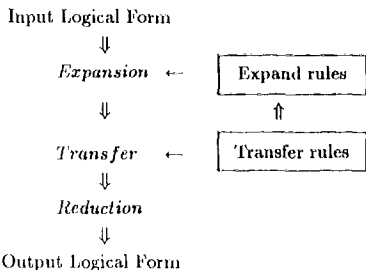


Figure 2: Transfer with Coordinate Expansion

In this form, *o* is a linguistic object, and a predicate *coord* represents a relation between the linguistic object and its constituents. The following is a result of expansion.

coord(w,w1)&wear(w1)&obj(w1,o1)&hat(o1)&
 coord(w,w2)&wear(w2)&obj(w2,o2)&shoe(o2)

The most naive and simplest strategy using this expansion is to expand every coordination within a sentence and represent it in sentence coordination before transfer. This transfer result will be reorganized again into an appropriate representation of coordination in the target language. But this solution seems inefficient from the computational point of view because cases where expansion is necessary are rare. Unnecessary expansion and reorganization of coordinate structures should be avoided.

The strategy we propose executes coordination expansion only if it is necessary [5]. Figure 2 shows a general view of our modified approach to transfer.

Transfer is divided into three phases; in the first phase, logical forms are expanded if expand rules (explained below) find the necessity of coordinate expansion. This process continues as long as the necessity remains. In the second phase, transfer described in previous sections is executed. Finally, in the third phase, coordination is reorganized if the target language has a more appropriate structure for coordination than the second phase result. (Consider the translation of (1b). Without reorganizing coordination, the transfer result will contain two “wear”s.)

The following is an expand rule which detects the necessity of expansion concerned with translation of “wearing a hat”.

wear(X)&obj(X,Y)&coord(Y,Y1)&HAT(Y1)&
 coord(Y,Y2)&Y1≠Y2&¬HAT(Y2)
 → expand(Y,X)

In the rule, *expand(Y,X)* means that coordination of the level Y should be expanded to that of X. This rule suggests that coordinate expansion is necessary

if the object of "wear" is a coordinate structure, and one constituent is a *HAT* while another isn't.

We assume that expand rules are produced automatically from transfer rules before the actual transfer operation. The simplest way of extracting such a rule refers only one transfer rule at one time. This means that the necessity of expansion can be detected not by creating expand rules but by referring transfer rules in actual transfer phase. But the former approach seems essential if we wish to optimize expansion detection.

5.2.2 Other examples

Expand rules from a transfer rule which involves a structural change are little different in forms to the previous case. The following are a transfer rule for translating "commit suicide" into Japanese and its expand rule.

commit(X)&obj(X,Y)&suicide(Y)
 → jisatsu_suru(X)

commit(X)&obj(X,Y)&coord(Y,Y1)&suicide(Y1)&
 coord(Y,Y2)&Y1≠Y2&¬suicide(Y2)
 → expand(Y,X)

Another example is the translation of the English "HUMAN have ADJ NOUN" construction into Japanese. The sentence (2E) is translated (2J) using the rule (3).

(2E) She has beautiful eyes.

(2J) kanojo-no me-ga utsukushii.
 she-poss eye-subj beautiful

(3) have(H)&subj(H,S)&obj(H,X)&mod(X,M)
 → poss(X,S)&subj(M,X)
 iff HUMAN(S)&PART_OF_BODY(X)

This case is more complicated than the previous ones because the transfer rule refers to two semantic features. Therefore we will get two expand rules, one of which is the following rule.

have(X)&subj(X,Y)&coord(Y,Y1)&HUMAN(Y1)&
 coord(Y,Y2)&Y1≠Y2&¬HUMAN(Y2) &
 obj(X,Z) & mod(Z,..) & PART_OF_BODY(Z)
 → expand(Y,X)

In addition, we need another expand rule which checks another type of coordinate construction. This rule will apply when the system translates "she has long hair and beautiful eyes".

have(X) & subj(X,S) & HUMAN(S) & obj(X,Y)
 & ¬mod(Y,..) & coord(Y,Y1) & mod(Y1,..) &
 PART_OF_BODY(Y1)
 → expand(Y,X)

5.2.3 Discussion

Our coordinate expansion detection works even if a coordinate structure has more than two constituents. What we have to consider is an appropriate expansion algorithm. For example, in translating (4a), an appropriate expansion should keep coordination of "shoes" and "stockings", as shown in (4b), because both satisfy a semantic constraint on which the system chooses "haku" as the translation of "wear". Otherwise reorganizing a coordination in the generation phase is inevitable.

(4a) She wears a hat, stockings and shoes.

(4b) She wears a hat and wears stockings and shoes.

Reorganization of a coordination in the target language does not only occur as outlined in the above case. Since the coordinate expansion is completely separate from the actual transfer process, transfer rules which do not cause problems might be used.

There is still a problem to be solved with regard to the transfer of coordination with "expansion"; expansion is not always straightforward. There is often a case where coordinate expansion is impossible without deep understanding of the sentences, or it is impossible in that it may change their semantic structures. For example, the sentence (5b) cannot be the expansion of (5a) though it seems so at first glance.

(5a) I saw a group of men and women.

(5b) I saw a group of men and I saw a group of women.

The apparent disadvantage of our approach with "expand rules" is that a large number of expansion rules might be created. Though it provides an efficient way of detecting the necessity of expansion, it consumes a lot of memory, which will raise an implementation problem.

6 Conclusion and future work

In this paper, we showed how complex structural changes in transfer are treated within our logic-based transfer model, in which the target linguistic description is obtained from the source description in a non-destructive manner. These type of structural changes cause a problem when they co-occur and their transfer rules are interacted. We also discussed a problem in transferring coordinate structures and presented an extended transfer model with coordinate expansion.

Some of our solutions to these problems are rather simple and need further investigation when we apply our framework to a practical system. We also have to evaluate our framework by examining a wider range of translation problems.

One of our current concerns is to implement our transfer model in a parallel processing framework. Our current algorithm for transfer and its implementation have a procedural aspect of operation. That

is, the sense of "logic" in the name of our model has not been fully achieved. We think that the search for parallel implementation will lead to "logic-based transfer" in the true sense.

References

- [1] H.Alshawi, D.Carter and M.Rayner: 'Translation by Quasi Logical Form Transfer', in *Proc. of 29th ACL Conference*, Berkeley, 1991.
- [2] C.Copeland, J.Durand, S.Krauwer and B.Maegaard: 'The Eurotra Linguistic Specification', in *Studies in Machine Translation and Natural Language Processing Vol. 1*, Office for Official Publication of the European Communities, 1991
- [3] R.Kaplan,K.Netter,J.Wedekind and A.Zaenan: 'Translations by Structural Correspondences', in *Proc. of 4th European ACL Conference*, Manchester, 1989.
- [4] S.Kinoshita and J.Tsuji: 'Framework for Incorporating Knowledge into Machine Translation', in *Proc. of International Workshop on Fundamental Research for the Future Generation of Natural Language Processing*, Kyoto, 1991.
- [5] S.Kinoshita: *Transfer of Coordinate Structures*, CCL/UMIST Report 91/13, 1991.
- [6] J.Lindop and J.Tsuji: *Complex Transfer in MT: A Survey of Examples*, CCL/UMIST Report 91/5, 1991.
- [7] S.Nirenburg: *Machine Translation*, Cambridge University Press, 1987.
- [8] J.D.Phillips, K.Yoshimura, K.Dohsaka and J. Tsujii: 'Linguistic Context for Machine Translation', in *Proc. of SICONLP '90*, Seoul, 1990.
- [9] J.D.Phillips: *An Architecture for Machine Translation*, CCL/UMIST Report 92/7, 1992.
- [10] S.Sato and M.Nagao: 'Memory-based Translation', in *Proc. of COLING 90*, Helsinki, 1990.
- [11] J.Tsuji and K.Fujita: 'Lexical Transfer based on Bilingual Signs: Towards interaction during transfer', in *Proc. of 5th European ACL Conference*, Berlin, 1991.
- [12] G.van Noord, J.Dorrepal et al: 'The MiMo2 Research System', in *Proc. of the 3rd International Conference on Theoretical and Methodological Issues in Machine Translation*, Austin, 1990.
- [13] H. Watanabe: 'A Model of a Transfer Process Using Combinations of Transfer Rules', in *Proc. of PRICAI '90*, 1990.