TRANSLATION BY QUASI LOGICAL FORM TRANSFER

Hiyan Alshawi, David Carter and Manny Rayner
SRI International
Cambridge Computer Science Research Centre
23 Millers Yard, Cambridge CB2 1RQ, U.K.
hiyan@cam.sri.com, dmc@cam.sri.com, manny@cam.sri.com

Björn Gambäck
Swedish Institute of Computer Science
Box 1263, S – 164 28 KISTA, Stockholm
gam@sics.se

ABSTRACT

The paper describes work on applying a general purpose natural language processing system to transfer-based interactive translation. Transfer takes place at the level of Quasi Logical Form (QLF), a contextually sensitive logical form representation which is deep enough for dealing with cross-linguistic differences. Theoretical arguments and experimental results are presented to support the claim that this framework has good properties in terms of modularity, compositionality, reversibility and monotonicity.

1 INTRODUCTION

In this paper we describe a translation project whose aim is to build an experimental Bilingual Conversation Interpreter (BCI) which will allow communication through typed text between two monolingual humans using different languages (cf Miike *et al*, 1988). The choice of languages for the prototype system is English and Swedish. Input sentences are analysed by the Core Language Engine (CLE[1]) as far as the level of Quasi Logical Form (QLF; Alshawi, 1990), and then, instead of further ambiguity resolution, undergo transfer into another QLF having constants and predicates corresponding to word senses in the other language. The transfer rules used in this process correspond to a certain kind of meaning postulate. The CLE then generates an output sentence from the target language QLF, using the same linguistic data as is used for analysis of that language.

[1] The CLE is described in Alshawi (1991) which includes more detailed discussion of the BCI architecture in a chapter by the present authors.

QLFs were selected as the appropriate level for transfer because they are far enough removed from surface linguistic form to provide the flexibility required by cross-linguistic differences. On the other hand, the linguistic, unification-based processing involved in creating them can be carried out efficiently and without the need to reason about the domain or context; the QLF language has constructs for explicit representation of contextually sensitive aspects of interpretation.

When it is necessary, for correct translation, to resolve an ambiguity present at QLF level, the BCI system interacts with the source language user to make the necessary decision, asking for a choice between word sense paraphrases or between alternative partial bracketings of the sentence. There is thus a strong connection between our choice of a representation sensitive to context and the use of interaction to resolve context dependent ambiguities, but in this paper we concentrate on representational and transfer issues.

2 CLE REPRESENTATION LEVELS

In this section we explain how QLF fits into the overall architecture of the CLE and in section 3 we discuss the reasons for choosing it for interactive dialogue translation.

2.1 CLE Processing Phases

A coarse view of the CLE architecture is that it consists of a linguistic analysis phase followed by a contextual interpretation phase. The output of the first phase is a set of alternative QLF analyses of a sentence, while the output of the second is an RQLF (resolved QLF) representation of the interpretation of an utterance:

$$Sentence \xrightarrow{\text{linguistic analysis}} QLFs$$
$$QLFs \xrightarrow{\text{contextual interpretation}} RQLF.$$

Deriving a fairly conventional Logical Form (LF) from the RQLF is then a simple formal mapping which removes the information in the RQLF that is not concerned with truth conditions.

Linguistic analysis and contextual interpretation each consist of several subphases. For analysis these are: orthography, morphological analysis, syntactic analysis (parsing), and (compositional) semantic analysis. Apart from the first, these analysis subphases are based on the unification grammar paradigm, and they all use declarative bidirectional rules.

When the CLE is being used as an interface to a computerized information system (e.g. a database system), its purpose is to derive an LF representation giving the truth conditions of an utterance input by a user. The LF language is based on first order predicate logic extended with generalized quantifiers and some other higher order constructs (Alshawi and van Eijck, 1989). For example, in a context where *she* can refer to Mary Smith, and *one* to "a car", a possible LF for *She hired one* is:

```
quant(exists,C,[car1,C],
   quant(exists,E,[event,E],
      [past,[hire1,E,mary_smith,C]])).
```

This can be paraphrased as "There is a car C, and an event E such that, in the past, E is a hiring event by Mary Smith of C." In this notation, quantified formulae consist of a generalized quantifier, a variable, a restriction and a scope; square brackets are used for the application of predicates and operators to their arguments. To arrive at such LF representations, a number of intermediate levels of representation are produced by successive modular components.

Generation of linguistic expressions in the CLE takes place from QLFs (or from RQLFs by mapping them to suitable QLFs). Since the rules used during the analysis phase are declarative and bidirectional, these are also used for generation. To achieve computationally efficient analysis and generation, the rules are pre-compiled in different ways for application in the two directions. Generation uses the semantic-head driven algorithm (Shieber *et al*, 1990).

2.2 The QLF Language

The QLF representations produced for a sentence are neutral with respect to the choice of referents for pronouns and definite descriptions, and relations implied by compound nouns and ellipsis. They are also neutral with respect to other ambiguities corresponding to alternative scopings of quantifiers and operators and to the collective/distributive and referential/attributive distinctions. The QLF is thus the level of representation encoding the results of compositional linguistic analysis independently of contextually sensitive aspects of understanding. These aspects are addressed by the contextual interpretation phase which has the following subphases: quantifier scoping (Moran 1988), reference resolution (Alshawi 1990), and plausibility judgement.

The QLF language is a superset of the LF language containing additional expressions corresponding, for example, to unresolved anaphors. More specifically, there are two additional term constructs (anaphoric terms and quantified terms), and one additional formula construct (anaphoric formulae):

`a_term(`*Category*`,`*EntityVar*`,`*Restriction*`).`
`q_term(`*Category*`,`*EntityVar*`,`*Restriction*`).`
`a_form(`*Category*`,`*PredVar*`,`*Restriction*`).`

These QLF constructs contain syntactic and morphological information in the *Category* and logical (truth-conditional) information in the *Restriction*, itself a QLF formula binding the variable. A QLF from which the LF for *She hired one* could have been derived is:

```
[past,
  [hire,
   q_term(<t=quant,n=sing>,
          E,[event,E]),
   a_term(<t=ref,p=pro,l=she,n=sing>,
          Y,[female,Y]),
   q_term(<t=quant,n=sing>,
          C,a_form(<t=pred,l=one>,
                 P, [P,C]))]].
```

in which categories are shown as lists of feature-value specifications (the feature shown are `t` for QLF expression type, `n` for number, `p` for phrase type, and `l` for lexical information). The differences between the QLF shown here and the LF shown earlier are that the quantified terms have been scoped, the anaphoric term for *she* has been resolved to Mary Smith, and the anaphoric NP restriction implicit in *one* has been resolved using the predicate `car`.

The RQLF representation of an utterance includes all the information from the QLF, together with the resolutions of QLF constructs made during the contextual interpretation phase. For example, the referent of an `a_term` is unified with the `a_term` variable.

Some constraints on plausibility can be applied at the QLF level before a full interpretation has been derived. This is because most of the predicate-argument structure of an utterance has been determined at that point, allowing, in particular, the application of sortal constraints expected by predicates of their arguments. Sortal constraints cut down on structural (e.g. attachment) ambiguity, and on word sense ambiguity, the latter being particularly important for the translation application in the context of large vocabularies.

3 REPRESENTATION LEVELS FOR TRANSFER

The representational structures on which transfer operates must contain information corresponding to several linguistic levels, including syntax and semantics. For transfer to be general, it must operate recursively on input representations. We call the level of representation on which this recursion operates the "organizing" level; semantic structure is the natural choice, since the basic requirement of translation is that it preserves meaning.

Syntactic phrase structure transfer, or deep-syntax transfer (e.g. Thurmair 1990, Nagao and Tsujii 1986) results in complex transfer rules, and the predicate-argument structure which is required for the application of sortal restrictions is not represented.

McCord's (1988, 1989) organizing level appears to be that of surface syntax, with additional deep syntactic and semantic content attached to nodes. As we have argued, this level is not optimal, which may be related to the fact that McCord's system is explicitly not symmetrical: different grammars are used for the analysis and synthesis of the same language, which are viewed as quite different tasks. Isabelle and Macklovitch (1986) argue against such asymmetry between analysis and synthesis on the grounds that, although it is tempting as a short-cut to building a structure sufficiently well-specified for synthesis to take place, asymmetry means that the transfer component must contain a lot of knowledge about the target language, with dire consequences for the modularity of the system and the reusability of different parts of it. In the BCI, however, the transfer rules contain only cross-linguistic knowledge, allowing the analysis and generation to make use of exactly the same data.

Kaplan *et al* (1989) allow multiple levels of representation to take part in the transfer relation. However, Sadler *et al* (1990) point out that the particular approach to realizing this taken by Kaplan *et al* has problems of its own and does not cleanly separate monolingual from contrastive knowledge.

The CLE processing subphases offer three semantic representations of different depth as candidates for an appropriate transfer level, namely QLF, RQLF and LF. At the LF level, sortal restrictions can be applied, but the form of noun phrase descriptions used and also information on topicalization is no longer present; the LF representation is too abstract for transfer. On the other hand, not all the information appearing in the RQLF about how QLF constructs have been resolved is necessary for translation. Resolved referents are not an adequate generator input for definite descriptions in the target language, since the view of the referent in the source is lost during translation. Another case is that translation from resolved ellipsis can result in unwieldy target sentences. In arguing for QLF-level transfer, we are asserting that predicate-argument relations of the type used in QLF are the appropriate organizing level for compositional transfer, while not denying the need for syntactic information to ensure that, for example, topichood or the given/new distinction is preserved.

Finally, in contrast to systems such as Rosetta (Landsbergen, 1986) which depend on stating rule by rule correspondences between source and target grammars, we wish to make the monolingual descriptions as independent as possible from the task of translating between two languages. Apart from its attractions from a theoretical point of view, this has practical advantages in allowing grammars to be reused for different language pairs and for applications other than translation.

4 QLF TRANSFER

QLF transfer involves taking a QLF analysis of a source sentence, say QLF1, and deriving from it another expression, QLF2, from which it is possible to generate a sentence in the target language. Leaving aside unresolved referential expressions, the main difference between QLF1 and QLF2 is that they will contain constants, particularly predicate constants, that originate in word sense entries from the lexicons of the respective languages. If more than one candidate source language QLF exists, the appropriate one is selected by presenting the user with choices of word sense paraphrases and of bracketings relating to differences in the syntactic analyses from which the QLFs were derived.

A transfer rule specifies a pair of QLF patterns. The left hand side matches QLF expressions for one language and the right hand side matches those for the other:

```
trans(<QLF1 subexpression pattern>
              <Operator>
      <QLF2 subexpression pattern>).
```

If the operator is == then the rule is bidirectional. Otherwise, a single direction of applicability is indicated by use of one of the operators >= or =<. Transfer rules are applied recursively, this process following the recursive structure of the source QLF. In order to allow transfer between structurally different QLFs, rules with 'transfer variables' need to be used. These variables, which take the form **tr(*atom*)**, show how subexpressions in the source QLF correspond to subexpressions translating them in the target QLF. For example, the following rule expresses an equivalence between the English *to be called* ("*I am called John*"), and the Swedish *heta* ("*Jag heter John*").

```
trans([call_name,
       tr(ev),
       q_term(<t=quant,n=sing>,
              A,[entity,A]),
       tr(ag),
       tr(name)]
      ==
      [heta1,tr(ev),tr(ag),tr(name)]).
```

Transfer rules often correspond directly to interlingual meaning postulates: when the expressions in a transfer rule are formulae, the symbols ==, >=, and =< can be read as the logical operators <-->, -->, and <-- respectively. A rule like

```
trans([and,[bad1,X],[luck1,X]]
      ==
      [otur1,X])
```

translating between the English *bad luck* and the Swedish *otur*, can be interpreted in this way.

We will now assess the method's strengths and weaknesses, as they have manifested themselves in practice. We will pay particular attention to the criteria of expressiveness, compositionality, simplicity, reversibility and monotonicity.

We take the last point first, since it is the most straightforward one. Since rules are applied purely nondeterministically and by pure unification, we get monotonicity "for free" – although there is a case for disallowing transfer by decomposition of a complex QLF structure which directly matches one side of a transfer rule. The other points need more discussion.

4.1 Expressiveness

Since we are intentionally limiting ourselves by not allowing access to full syntactic information (but only to that placed in QLF categories) in the transfer phase, it is legitimate to wonder whether the formalism can really be sufficiently expressive. Here, we will attempt to answer this criticism; we begin by noting that shortcomings in this area can be of several distinct kinds. Sometimes, a formalism can appear to make it necessary to write many rules, where one feels intuitively that one should be enough; we treat this kind of problem under the heading of compositionality. In other cases, the difficulty is rather that there does not appear to be any way of expressing the rule at all in terms of the given formalism. In our case, a fair proportion of problems that at first seem to fall into this category can be eliminated by having adequate monolingual grammars and using the target grammar as a filter; the idea is to allow the transfer component to produce unacceptable QLFs which are filtered out by fully constrained target grammars.

A good example of the use of this technique is the English definite article, which in Swedish can be translated as a gender-dependent article, but preferably is omitted; however, an article is obligatory before an adjective. Solving this problem

Table 1: *Types of complex transfer used*	
Type	Example
Different particles	John **likes** Mary John **tycker om** Mary
Passive to active	Insurance **is included** Försäkring **ingår**
Verb to adjective	John **owes** Mary $20 John **är skyldig** Mary $20
Support verb to normal verb	John **had** an accident John **råkade ut för** en olycka
Single verb to phrase	John **wants** a car John **vill ha** en bil (lit.: "wants to have")
Idiomatic use of PP	John **is in a hurry** John **har bråttom** (lit.: "has hurry")

at transfer level is not possible, since the transfer component has no way of knowing that a piece of logical form will be realized as an adjective; there are many cases where an adjective-noun combination in English is best translated as a compound noun in Swedish. Exploiting the fact that the relevant constraint is present in the Swedish grammar, however, the "transfer-and-filter" method reduces the problem to two simple lexical rules. Sortal restrictions at the target end can also be used as a filter in a similar way.

4.2 Simplicity and reversibility

The most obvious way to put the case with regard to simplicity is by giving a count of the various categories of rule, and providing evidence that there is a substantial proportion of rules which are simple in our framework, but would not necessarily be so in others.

The transfer component currently contains 718 rules. 576 of these (80.2%) have the property that both the right- and left-hand sides are atomic. 502 members of this first group (69.9%) translate senses of single words to senses of single words; the remaining 74 (10.3%) translate atomic constants representing the senses of complex syntactic constructions, most commonly verbs taking particles, reflexives, or complementizers. An example is the following rule, which defines an equivalence between English *care about* (*"John cares about Mary"*) and Swedish *bry sig om* (*"John bryr sig om Mary"*, lit. "John cares himself about Mary").

Table 2: *Transfer contexts used*	
Context	Example
Perfect tense	John has liked Mary John har tyckt om Mary
Negated	John doesn't like Mary John tycker inte om Mary
YN-question	Does John like Mary? Tycker John om Mary?
WH-question	Who does John like? Vem tycker John om?
Passive	Mary was liked by John Mary blev omtyckt av John
Relative clause	The woman that John likes Kvinnan som John tycker om
Sentential complement	I think John likes Mary Jag tror John tycker om Mary
Embedded question	I know who John likes Jag vet vem John tycker om
VP modifier	John likes Mary today John tycker om Mary idag
Object raising	I want John to like Mary Jag vill att John ska tycka om Mary ("I want that J. shall like M.")
Change of aspect	John stopped liking Mary John slutade tycka om Mary ("J. stopped like-INF M.")

```
trans(care_about == bry_sig_om).
```

Since vocabulary has primarily been selected with regard to utility (we have, for example, made considerable use of frequency dictionaries (Allén 1970)), we think it reasonable to claim that QLF-based transfer is simplifying the construction of transfer rules in a substantial proportion of the commonly encountered cases.

On the score of reversibility, we will once again count cases; here we find that 659 (91.8%) of the rules are reversible, 17 (2.4%) work only in the English-Swedish direction, and 42 (5.8%) only in the Swedish-English direction. These also seem to be fairly good figures.

4.3 Compositionality

As in any rule-based system, "compositionality" corresponds to the extent to which it is necessary to provide special mechanisms to cover cases of irregular interactions between rules. As far as we know, there is no accepted benchmark for testing compositionality of transfer; what we have done, as a first step in this direction, is to select six common types of complex transfer, and eleven common contexts in which they can occur. These are summarized in tables 1 and 2 respectively. Each complex transfer type is represented by a sample rule, as shown in table 1; the question is the extent to which the complex transfer rules continue to function in the different contexts (table 2).

To test transfer compositionality properly, it is not sufficient simply to note which rule/context combinations are handled correctly; after all, it is always possible to create a completely *ad hoc* solution by simply adding one transfer rule for each combination. The problem must rather be posed in the following terms: if there is a single rule for each complex transfer type, and a number of rules for each context, how many *extra* rules must be added to cover special combinations? It is this issue we will address.

The actual results of the tests were as follows. There were 124 meaningful combinations (some constructions could not be passivized); in 103 of these, transfer was perfectly compositional, and no extra rule was needed. For example, the English sentence for the combination "Verb to adjective + WH-question" is *How much does John owe Mary.* The corresponding Swedish sentence is *Hur mycket är John skyldig Mary?* ("How much is John indebted-to Mary?"), and the two QLFs are[2]:

```
[whq,
 [pres,
  [owe_have_to_pay,
   q_term(<t=quant,n=sing>,A,[event,A]),
   a_term(<t=ref,p=name>,
          B,[name_of,B,john]),
   q_term(<t=quant,l=wh>,C,[quantity,C]),
   a_term(<t=ref,p=name>,
          D,[name_of,D,mary])]]]

[whq,
 [present,
  [vara,
   q_term(<t=quant,n=sing>,A,[state,A]),
   [skyldig_ngn_ngt,
    a_term(<t=ref,p=name>,
           B,[name_of,B,john]),
    a_term(<t=ref,p=name>,
           C,[name_of,C,mary]),
    q_term(<t=quant,l=wh>,D,[quantity,D])]]]]
```

[2] *är* is the present tense of *vara*.

It should be evident that the complex transfer rule defining the equivalence between *owe* and *vara skyldig*,

```
trans([owe_have_to_pay,
       q_term(<t=quant,n=sing>,A,[event,A]),
       tr(ag),tr(sum),tr(obj)]
      ==
      [vara,
       q_term(<t=quant,n=sing>,A,[state,A]),
       [skyldig_ngn_ngt,
        tr(ag),tr(obj),tr(sum)]]).
```

is quite unaffected by being used in the context of a WH-question.

Of the remaining 21 rule/context/direction triples, seven failed for basically uninteresting reasons: the combination "Perfect tense + Passive-to-active" did not generate in English, and the six sentences with the object-raising rule all failed in the Swedish-English direction due to the transfer component's current inability to create a function-application from a closed form. The final fourteen failures are significant from our point of view, and it is interesting to note that all of them resulted from mismatches in the scope of tense and negation operators.

The question now becomes that of ascertaining the generality of the extra rules that need to be added to solve these fourteen unwanted interactions. Analysis showed that it was possible to add 26 extra rules (two of which were relevant here), which reordered the scopes of tense, negation and modifiers, and accounted for the scope differences between the English and Swedish QLFs arising from the general divergences in word-order and negation of main verbs. These solved ten of the outstanding cases. For example, the combination "Different particles + Negated" is *John doesn't like Mary* in English and *John tycker inte om Mary* (lit.: "John thinks not about Mary") in Swedish; the QLF-pair is:

```
[pres,
 [not,
  [like,
   q_term(<t=quant,n=sing>,A,[event,A]),
   a_term(<t=ref,p=name>,
          B,[name_of,B,john]),
   a_term(<t=ref,p=name>,
          B,[name_of,B,mary])]]]
```

```
[not,
 [present,
  [tycka_om,
   q_term(<t=quant,n=sing>,A,[event,A]),
   a_term(<t=ref,p=name>,
          B,[name_of,B,john]),
   a_term(<t=ref,p=name>,
          B,[name_of,B,mary])]]]
```

The extra rule here,

```
trans([pres,[not,tr(body)]] ==
      [not,[present,tr(body)]]).
```

reorders the scopes of the negation and present-tense operators, but does not need to access the interior structure of the QLF (the "body" variable); this turns out to be the case for most interactions of negation, VP-modification and complex transfer. It is thus not surprising that a small number of similar rules covers most of the cases.

The four bad interactions left all involved the English verb *to be*; these were the combinations "Passive to active + VP modifier" and "Idiomatic use of PP + negation", which failed to transfer in either direction. Here, there is no general solution involving the addition of a small number of extra rules, since the problem is caused by an occurrence of *to be* on the English side that is not matched by an occurrence of the corresponding Swedish word on the other. The solution must rather be to add an extra rule *for each complex transfer rule in the relevant class* to cover the bad interaction. To solve the specific examples in the test set, two extra rules were thus required.

Summarizing the picture, the tests revealed that all bad interactions between the transfer rules and contexts shown here could be removed by adding four extra rules to cover the 124 possible interactions. In a general perspective (viewing the rules as representatives of their respective classes), the rule-interaction problems exemplified by the concrete collisions were solved by adding

- 26 general rules to cover certain standard scope mismatches caused by verb-inversion and negation.
- two extra rules (one for present and one for past tense) for each complex transfer rule of either the "Idiomatic use of PP" or "Active to Passive" types, to cover idiosyncratic interactions of these with negation and VP-modification respectively.

We view these results as very promising: there were few bad interactions, and those that existed were of a regular nature that could be counteracted without fear of further unwelcome side-effects. This gives good grounds for hoping that the system could be scaled up to a practically useful size without suffering the usual fate of drowning in a sea of *ad hoc* fixes.

5 IMPLEMENTATION STATUS

The current implementation includes analysis, transfer, and generation modules, sizable grammars with morphological, syntactic and semantic rules for English and Swedish, and an experimental set of transfer rules for this language pair. Relative to the size of the grammars, the lexicons are still small (approximately 2000 and 1000 words respectively). About 250 entries for each language have been added for a specific domain (car hire), which makes possible moderately unconstrained conversation on this topic; the system, including the facilities for interactive resolution of translation problems, has been tested on a corpus of about 400 sentences relating to the domain. For short sentences typical of the car hire domain, median total processing times for analysis, transfer and generation are around ten seconds when running under Quintus Prolog on a SUN SPARCstation 2.

We are currently investigating a different QLF representation of tense, aspect and modality which should increase the transfer compositionality for the operator cases we have discussed in this paper, as well as allowing more flexible resolution of temporal relations in applications other than translation.

ACKNOWLEDGMENTS

The work reported here was funded by the Swedish Institute of Computer Science, and the greater part of it was carried out while the third author was employed there. We would like to thank Steve Pulman for many helpful discussions, especially with regard to the problems encountered in adapting the English grammar to Swedish.

REFERENCES

Allén, Sture (ed.) (1970) *Frequency Dictionary of Present-Day Swedish*, Almqvist & Wiksell, Stockholm.

Alshawi, Hiyan and Jan van Eijck (1989) "Logical Forms in the Core Language Engine". *27th Annual Meeting of the Association for Computational Linguistics*, Vancouver, British Columbia, pp. 25–32.

Alshawi, Hiyan (1990) "Resolving Quasi Logical Forms". *Computational Linguistics*, Vol. 16, pp. 133-144.

Alshawi, Hiyan, ed. (to appear 1991). *The Core Language Engine*. Cambridge, Massachusetts: The MIT Press.

Kaplan, Ronald M., Klaus Netter, Jürgen Wedekind and Annie Zaenen (1989) "Translation by Structural Correspondences", *Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, pp. 272–281.

Isabelle, Pierre and Elliot Macklovitch (1986) "Transfer and MT Modularity", *Eleventh International Conference on Computational Linguistics* (COLING-86), Bonn, pp. 115–117.

Landsbergen, Jan (1986) "Isomorphic grammars and their use in the Rosetta translation system", in M. King (ed), *Machine Translation Today: the State of the Art*, Edinburgh University Press, Edinburgh.

McCord, Michael C. (1988) "A Multi-Target Machine Translation System", *Proceedings of the International Conference on Fifth Generation Computer Systems*, Tokyo, pp. 1141–1149.

McCord, Michael C. (1989) "Design of LMT: a Prolog-based Machine Translation System", *Computational Linguistics*, Vol. 15, pp. 33–52.

Miike, Seiji, Koichi Hasebe, Harold Somers, and Shin-ya Amano (1988) "Experiences with an on-line translating dialogue system", *26th Annual Meeting of the Association for Computational Linguistics*, State University of New York at Buffalo, Buffalo, New York, pp. 155–162.

Moran, Douglas B. (1988). "Quantifier Scoping in the SRI Core Language Engine", *26th Annual Meeting of the Association for Computational Linguistics*, State University of New York at Buffalo, New York, pp. 33–40.

Nagao, Makoto, and Jun-ichi Tsujii (1986) "The Transfer Phase of the Mu Machine Translation System", *Eleventh International Conference on Computational Linguistics* (COLING-86), Bonn, pp. 97–103.

Sadler, Louisa, Ian Crookston, Douglas Arnold and Andrew Way (1990) "LFG and Translation", *Third International Conference on Theoretical and Methodological Issues in Machine Translation*, Linguistics Research Center, Austin, Texas.

Shieber, Stuart M., Gertjan van Noord, Fernando C.N. Pereira and Robert C. Moore (1990) "Semantic-Head-Driven Generation", *Computational Linguistics*, Vol. 16, pp. 30-43.

Thurmair, Gregor (1990) "Complex lexical transfer in METAL", *Third International Conference on Theoretical and Methodological Issues in Machine Translation*, Linguistics Research Center, Austin, Texas.