A New Formalism for Lexical Transfer: Bilingual Sign-Based Formalism*

Jun-ichi Tsujii & Kimikazu Fujita

The lexical tranfer has been one of major sources of difficulties in MT. Apart from difficulties of selecting appropriate target equivalent words, systematic structural correspondences between two languages are often violated by idiosyncracies of lexical correspondences. We can hardly expect to have systematic theories of lexical tranfer which generalize idiosyncracies in lexical correspondences. What we can do is to prepare a framework in which we can accumulate systematically idiosyncratic knowledge concerned with lexical correspondences. In this paper, we propose a framework based on bilingual signs for lexical transfer. Through definitions of bilingual signs, not only local linguistic structures of two languages but also predicates which are used for specifying extra-linguistic knowledge are related with each other. Furthermore, by using explicit specifications of mutual relationships between bilingual signs, the system can formulate meaningful questions for disambiguating "transfer ambiguities" during lexical transfer. Compared with converntional bilingual dictionaries, our proposed framework has several desirable properties such as declarativeness, reversibility and flexible interface with extralinguistic processings. A rough sketch of the transfer phase which is based on constraint relaxation is also given.

1. Introduction

'Lexical Transfer' has always been one of the main sources of problems

* This is a paper originally presented at the Seoul International Conference on Natural Language Processing on November 22, 1990.

in Machine Translation(MT) (Melby 1986) (Nirenburg 1988).

Research in transfer-based MT system has focused on discovering an appropriate level of linguistic description for translation, at which we can specify 'translation relations' (or transfer rules) in a simple manner. In the extreme, the Eurotra project has introduced the principle of 'simple transfer' which means that most parts of the descriptions of source language objects, especially their geometrical structures, should be preserved in the target language during transfer except for the lexical items. However, structural mismatches between two languages are frequently caused by lexical differences so that the effort of minimizing structural changes with 'lexical transfer' untouched seems to have caused serious difficulties.

Besides structural changes caused by lexical transfer, selecting equivalent target words for source lexical items has been one of the hardest problems in MT. A single source lexical item should be translated into different target expressions (words or phrases), depending on the linguistic or extralinguistic contexts in which it actually appears. Determining appropriate target equivalents for source words is one of the typical processes which require 'inferences' based on 'extra-linguistic' knowledge and/or information conveyed by other parts of the text.

Because individual languages have their own ways of reflecting the structure of the world in their lexicons, and the process of lexicalization in language is more or less arbitrary, we can hardly expect to have general theories of lexical correspondences across languages. We have to accumulate idiosyncratic 'knowledge' about individual lexical correspondences as bilingual knowledge. The bilingual knowledge used in the transfer phase of actual MT systems mostly consists of such idiosyncratic lexical correspondences.

What sort of information is necessary to select appropriate target equivalents and what structural changes are caused by 'lexical transfer' are highly dependent on language pairs and individual words. What we need is to prepare a framework in which such bilingual knowledge about idiosyncratic 'lexical correspondences' can be described in a natural way.

Our approach in this paper follows the general trend in computational linguistics which emphasizes the role of the lexicon in linguistic theories, and in particular, it shares the idea with Whitelock(1988) who proposed specifying most bilingual correspondences in lexicon.

Unlike former approaches, however, we explicitly introduce bilingual signs(e.g. 'signs' defined by words - monolingual signs - of two languages), and lexical descriptions which play the central role in transfer are given to these signs. Bilingual signs in our framework not only link the local linguistic structures of two languages where the corresponding two monolingual signs appear, but also, by behaving as a logical predicate, they connect linguistic-based processes in MT with inference processes. We also show that complicated structural changes which are often required in translation of remote language pairs like English and Japanese can be captured as special types of logical inferences.

The framework has the following advantages over conventional methods.

(i) Reversibility of bilingual dictionaries
(ii) Natural interfaces between knowledge-based (inference) processes with MT
(iii) Ease of paraphrasing using different words

We are now engaged in developing an MT system for monolingual users, who do not have enough knowledge about the target language (Tsujii, J. & M. Nagao(1988)). In such an environment, it is crucial for the system to pose appropriate questions to users in order to disambiguate ambiguities. In particular, to formulate appropriate questions for resolving ambiguities during the transfer phase is important and much more difficult, if we compared them with ambiguities in the analysis phase. We have to formulate questions which make differences in readings explicit, the differences which monolingual speakers of the source language usually do not recognize. The advantage (iii) of our framework will play an important role in this question formation process.

2. Bilingual Signs as Logical Predicates

In a naive formal semantics(FS), the meaning of a word, say *run*, is expressed by a predicate (say, RUN) which corresponds straightforwardly to the surface word. In short, based on this correspondence of basic units in the two domains, i.e. words in the linguistic domain and predicates in the domain of logic, FS has developed theories of relating complex expressions of the two domains which consist of more than one basic unit.

The (truth-conditional) meaning of a sentence (1) may be expressed, for example, as (2)

(1) The teacher runs the program.

(2) (exist e) (exist χ) (exist y)
$\{$RUN(e) & ARG1(e, χ) & ARG2(e, y)
TEACHER(χ) & PROGRAM(y)$\}$

We need more sophisticated notations for expressing the meanings of articles, quantifers, etc., and also in MT, we have to preserve other kinds of information which are conveyed by linguistic forms but can hardly be expressed in logical formulae (topic, focus, etc.). Actually, we are thinking of descriptions like QLF(quasi-logical form) (Alshawi (1989)) in which such information relevant to MT is encoded and annotated in logical formulae. However, for the sake of simplicity, we will ignore these required sophistications and use throughout this paper much simplified notations such as (2). We will also omit quantifiers in the following which are irrelevant to the current discussion. But even in more sophisticated representations such as QLF, the fact that the meanings of *teacher*, *run* and *program* are expressed straightforwardly by the predicates like TEACHER, RUN, and PROGRAM is untouched.

The meaning representation as such is purely monolingual. The truth condition of being *teacher* is assumed to be expressed by the predicate TEACHER.

The framework which we need for translation is essentially similar to the ones in FS, i.e. a framework of systematically relating complex expressions of two infinite domains, the domains of expressions of the source and target languages. However, as we said, the naive assumption in FS of straightforward correspondences of basic units does not hold in our domains.

Run, for example, has to be translated into several different Japanese verbs. While *run* in (1) is translated as *jikkoosuru* in Japanse, *run* in (3) should be translated differently (in this case, *un'eisuru).*

(3) The teacher runs the big company.

One can claim that *run* is monolingually ambiguous and that the above two senses should be distinguished by using different predicates, say, RUN1 and RUN2. This approach leads us to the well-known difficulty that the distinction of senses of single lexical items is, except for obvious cases, notoriously subjective and subjective judgements given by lexicographers have caused serious problems in actual MT systems such as difficulties in maintaining consistency between monolingual and bilingual dictionaries, etc. Furthermore, the required fineness of distinction of word senses depends highly on the target language (source words are translationally ambiguous (Tsujii (1986)).

The basic idea of bilingual signs is very simple. Instead of using predicates corresponding directly to surface words like RUN or predicates corresponding to intuitively distinguished senses like RUN1 and RUN2, we use pairs of lexical items of two languages as predicates. That is, we use [RUN:JIKKOOSURU], [RUN:UN'EISURU], etc. as basic predicates to express meanings of sentences.

The bilingual sign [RUN:JIKKOOSURU] is a predicate which expresses the truth condition an event *e* should satisfy in order to be described by *run* in English and *jikkoosuru* in Japanese. Note that [RUN:JIKKOOSURU] expresses not only one disambiguated sense of *run* but also one disambiguated sense of the Japanese verb *jikkoosuru* (*jikkoosuru* can be translated into several English verbs including *run, carry out, execute, implement, practice,* etc., depending on the context). [RUN:JIKKOOSURU] expresses the conjunction of truth conditions of *run* and *jikkoosuru.*

Though each bilingual sign basically corresponds to a lexical transfer rule (of the most simple sorts like *run* ↔ *un'eisuru, run* ↔ *jikkoosuru,* etc.; we will extend the bilingual signs to treat more complex lexical transfer in section 4), we give them independent status as predicates. We can use them to express truth conditional meanings of source and target linguistic forms and also to represent extralinguistic knowledge about the world (domain knowledge).

As Sadler(1990) pointed out, compared with the methods using arbitrary predicates (or units of frames, if you like), this method is well-motivated in selecting basic predicates (or frames). We can expect the set of bilingually defined predicates to have appropriate granularity for representing knowledge for translation, at least a necessary, if not sufficient, level of granularity for translation of two given languages.

Furthermore, we can use logical formulae to specify mutual relationships among bilingual signs, which means that we can specify explicitly 'logical' relationships among lexical tranfer rules. As we discuss later, this explicitness of relationships among 'lexical transfer rules' enhances the reversibility of rules and also provides the system with the capability of paraphrasing, which is crucial for disambiguation by dialogue in the transfer phase.

3. Definition of Bilingual Signs

We assume here a conventional transfer-based MT system where the monolingual analysis and transfer phases are executed separately by using separate monolingual and bilingual dictionaries (for more detailed discussion on the interface of these two stages, see Section 7). The analysis phase of English produces the following schema of logical formula (4) as the description of (1), instead of the logical formula (2).

(4) {[RUN:?1](e) & ARG1(e, χ) & ARG2(e, y)
& [TEACHER:?2](χ) & [PROGRAM:?3](y)}

The formula in (4) is not a logical formula but a schema which represents set of possible formulae. [RUN:?1] is a predicate schema, and by binding the variable '?1' to a specific Japanese verb, we get a specific predicate such as [RUN:JIKKOOSURU], [RUN:UN'EISURU], etc. The transfer phase is taken to be a phase which identifies appropriate predicates in a schema of logical formulae produced by the analysis phase.

As in conventional transfer-based MT systems, we use sortal restrictions on arguments for disambiguation, e.g. for identification of appropriate (bilingual) predicates. That is, predicates have their own arities and their sortal restrictions on the arguments.

[RUN:UN'EISURU], for example, requires the argument-1 to be a human or an organization and the argument-2 to be an organization. These restrictions can be expressed simply by logical implications such as (5) and (6) :

(5) ($\rightarrow$ [RUN:UN'EISURU](e) & ARG1(e,χ)
[ORGANIZATION:SOSHIKI](χ)v [HUMAN:NINGEN](χ))

(6) (→ [RUN:UN'EISURU](e) & ARG2(e,χ)
[ORGANIZATION:SOSHIKI](χ)).

These formulae mean that if the event e satisfies the predicate [RUN:UN' EISURU], then the object related with the event by ARG2 (we call this object the argument-2 of the event) should satisfy the predicate [ORGANIZATION:SOSHIKI], etc.

Because the event e is described in English by using *run*, it should satisfy one of the predicates of the predicate schema [RUN:?1]. Assume that it satisfies [RUN:UN'EISURU], then the argument-2 of the event should satisfy [ORGANIZATION:SOSHIKI]. If the sort of the actual argument contradicts this constraint, then the event does not belong to the sort [RUN: UN'EISURU] and it cannot be described by the Japanese verb *un'eisuru*. The lexical transfer *run* to *un'eisuru* is judged not appropriate in this case.

Logical implications are used to express various, ontologically different relationships between two formulae, which should be treated differently in MT. The consequence parts of the implications in the above examples express constraints which an event e and the participants of the event (arguments) should satisfy to be describable by both *run* and *un'eisuru*, e.g. the constraints and event belonging to the sort [RUN:UN'EISURU] should satisfy.

The other type of relation expressed by logical implications is a hierarchical relationship among sorts such as '[TEACHER:SENSEI] is a [HUMAN:NINGEN]', etc. This relationship plays an important role in translation, because two predicates linked by this relation can be used to describe the same 'event' or 'object' though the predicate of the supersort describes it in a more vague way than the predicate of the subsort. That is, 'human' can be used to describe the same object as 'teacher', but the description using 'human' is less accurate (more vague) than that using 'teacher', As we will discuss later, our trasfer phase tries to find target expressions which have appropriate levels of accuracy (vagueness).

These two types of implication are distinguished by the first argument of '→' as follows, beacuse they are used differently in translation (for more detailed notation, see Section 5).

(6') (→C [RUN:UN'EISURU] (e) & ARG2(e,χ)
[ORGANIZATION:SOSHIKI](χ))

(7) (→S [TEACHER:SENSEI] (χ)
[HUNAN:NINGEN](χ)).

The consequent parts of C-type implications like (6′) can be arbitrarily complex. However, because it has been already shown (Tsujii, (1986)) that simple sortal restrictions on arguments are often useful for disambiguating transfer ambiguities and also that the checking of sortal consistency based on sort-hierachies defined by simple S-type implication like (7) can be implemented in a computationally efficient manner, we introduce notational conventions in the definition part of bilingual signs to express sortal restrictions on arguments.

(8) (Def-Pred
[RUN:UN'EISURU]
{arg1 : =[HUMAN:NINGEN] v [ORGANIZATION:SOSHIKI],
arg2 : =[ORGANIZATION:SOSHIKI]})

(9) (Def-Pred
[RUN:JIKKOOSURU]
{arg1 : =[HUMAN:NINGEN],
arg2 :=[PROGRAM:PUROGURAMU]})

The definition (8) shows that the predicate [RUN:UN'EISURU] has the arity two and the arguments have sortal restrictions expressed by (5) and (6) (or (6′)). This definition is read also as a definition of an event sort [RUN:UN'EISURU]. Sorts of events are also organized into a hierarchy. We assume that objects/events are organized in separate hierarchies and that both hierarchies constitute lattices (we allow a sort to have multiple super-sorts).

From (4) (the schema of logical formulae for (1)), we know that argument-2 of the event e is filled by the variable y and that y satisfies some of the predicates in the predicate schema [PROGRAM:?3] (i.e. y belongs to some sorts in the schema [PROGRAM:?3]). If a sort-hierarchy shows that all the sorts in [PROGRAM:?3] are inconsistent with [ORGANIZATION: SOSHIKI], then the attempt of describing the event e by *un'eisuru* in Japanese fails and the transfer (*run* → *un'eisuru*) is rejected. On the other hand, the variables χ *(teacher)* and y *(program)* in the schema (4) may satisfy the sortal restrictions of [RUN:JIKKOOSURU] in (9) so that the transfer (*run* → *jikkoosuru*) is taken as feasible.

As in LFG, we assume that semantic representations (logical forms) are related lexically with a certain level of description of linguistic forms. Though LFG specifies the relations between functional structures and meaning representations (logical forms) in the monolingual lexicon of surface words, we specify them in bilingual lexicon, that is, in the 'lexical' descriptions of bilingual signs (predicates).

The relation of grammatical functions and predicate agruments is specified in LFG by the following description.

```
(10) run:
        sub :=
        obj :=
        sem :=[RUN 〈! sub〉 〈! obj〉]
```

In our framework, the specification of the relationships is done the other way around. Note also that: because a bilingual sign (predicate) is defined by two languages (English and Japanese), the two relationships of (logical form ↔ English) and (logical form ↔ Japanese) are specified in the same place.

```
(11) (Def-Pred
     [RUN:UN'EISURU]
     {arg1 = [HUMAN:NINGEN] v [ORGANIZATION:SOSHIKI},
      arg2 : = [ORGANIZATION:SOSHIKI],
      eng  : = {head : ={e-lex : = run},
            agt : = 〈! arg1〉,
            obj : = 〈! arg2〉},
      jpn  : = {head : = {j-lex : = un'eisuru},
            agt : = 〈! arg1〉,
            obj : = 〈! arg2〉}})
```

Here, in order to avoid further complications caused by changes of grammatical function (passivization, etc.), we use thematic role representations as linguistic descriptions. In the above, angle brackets '〈 〉' show a path description and exclamation-mark '!' in the angle brackets means the smallest description block (shown by braces '{ }') which contains the description block in which the '!' appears.

4. Complex Structural Changes-Complex Bilingual Signs

The schema (11) in Section 3 shows a simple case of a lexical transfer where the lexical correspondence (*run* ↔ *un'eisuru*) does not interact with the structure. The thematic role structures remain the same in the two languages. However, many cases have been observed where lexical transfer causes structural change. It is also often the case that single words in English correspond to phrasal expressions in Japanese, and vice versa.

We may expect that classes of objects or events which can be expressed by single words in one language correspond to natural (not contingent) classes of objects/events, the classes whose truth conditions are naturally captured by single predicates in logical forms. Therefore, we prepare single bilingual signs for expressing their truth conditions if at least one of the languages has lexical items, even though the other language lacks corresponding words and uses complex phrasal expressions to express the same classes of events/objects.

The following shows how our framework based on bilingual signs treats structural changes caused by lexical correspondences.

[A] Case changes : The English sentence (12a) is usually into (12b) in French.

(12a) I like him.
(12b) Il me plaît.

Though one may think of (monolingual) levels of linguistic descriptions where this discrepancy of syntactic functions disappears, it seems more natural to take this discrepancy as a bilingual phenomenon, i.e. the two languages have their own lexical items which capture the same states/events from their own viewpoints.

In our framework, corresponding case elements in two languages are linked with each other through the same argument names of bilingual signs so that this sort of case change can be treated in a simple way.

```
(13) (Def-Pred
       [LIKE:PLAIRE]
       {arg1 : =,
        arg2 : =,
```

```
eng  : = {head : = {e-lex : =like},
          agt : = ⟨! arg1⟩,
          obj : = ⟨! arg2⟩},
fre : = {head : = {f-lex : = plaire},
          agt : = ⟨! arg2⟩,
          obj : = ⟨! arg1⟩}})
```

[B] Lexical inclusions of arguments : A Japanese verb *nuru*, for example, is translated as *paint*, *varnish*, *spread* (bread with butter), *apply* (paint), etc., depending on the material. Some of the English verbs (*paint*, *varnish*, etc.) include the objects (in Japanese) in their meaning. For example,

(14a)	kabe-ni [n:wall -location]	peki-wo [n:paint -object]	nuru [v]
(14b)	(someone)	paints [v]	the wall [object]

This structural change is treated by the following definition of a bilingual sign [PAINT:PENKI-WO-NURU].

```
(15) (Def-Pred
     [PAINT:PENKI-WO-NURU]
     {arg1 :=,
      arg2 :=,
      eng  := {head := {e-lex :=paint-1}},
               agt :=⟨! arg1⟩,
               obj := ⟨! arg2⟩},
      jpn  := {head :={j-lex :=nuru},
               agt :=⟨! arg1⟩,
               obj := {head := {j-lex := penki}},
               loc :=⟨! arg2⟩}})
```

Note that Japanese verb *nuru* governs three dependents but one of them is in this definition filled in advance by a specific noun (*penki-paint* in English). The suffix '-1' in 'paint-1' is for distinguishing the verb usage of *paint* from its noun usages.

[C] Lexical gaps : Quite a number of English adjectives should be translated into clauses in Japanese, because Japanese simply lacks corresponding words. *Efficient* should be translated into a clause in Japanese as follows.

(16)	Kooritsu-ga	yoi
	[n:efficiency-subject]	[adj:good]

The phenomenon is similar to lexical inclusion except for the fact that this causes rather drastic changes in syntactic structures. In this paper, we assume that the descriptions on which transfer operations are performed are abstract enough so that the determination of syntactic categories, etc. is carried out in the generation phase.

```
(17) (Def-Pred
     [EFFICIENT:KOORITSU-GA-YOI]
     {arg1 :=,
      eng  := {head := {e-elx :=efficient},
               obj := ⟨! arg1⟩},
      jpn :=  {head := {j-lex := yoi},
               topic := ⟨! arg1⟩,
               obj := {head := {j-lex := kooritsu}}}})
```

[D] Category changes: One of the well-known examples of this is the correspondence between the English verb *like* and the Dutch adverb *graag* (which roughly corresponds to the English adverb *pleasantly*). The mental state expressed by the verb *like* is expressed not by a main verb but by an adverb in Dutch. This kind of phenomenon has also been observed very frequently in translation between English and Japanese, i.e. verbs in English which take sentential complements (especially infinitival clauses) are expressed specific adverbs modifying main clauses in Japanese which correspond to sentential complements in English.

The event expressed by the verb *manage* (in the usage of *manage to do something*) is captured by an adverb *nantoka* ('somehow or other', 'with great effort', etc.) in Japanese. The adverb is used to modify the event expressed in English as an infinitive clause(*to do something*).

(18a)	watashi-ga	nantoka	ronbun-wo	shiage	-ta.
	[n:I -subject]	[adv:some- how or other]	[n:paper -object]	[v:comp- plete]	[tense:past]

(18b) I managed to complete {the/a} paper.

This correspondence is captured by the following definition.

```
(19) (Def-Pred
      [MANAGE:NANTOKA]
      {arg1 :=,
       arg2 := [event:dekigoto],
       eng  := {head := {e-lex := manage],
                agt := <! arg1>,
                evt := <! arg2>},
       jpn := {<! arg2>,
                agt := <! arg1>,
                /adv := {head := {j-lex := nantoka}}}})
```

In this example, though the adverb *nantoka* is not the head (semantic governer) of the Japanese deep case description ('jpn'), it is converted into the predicate [MANAGE:NANTOKA] in the logical formula, and the rest of the 'jpn' description into the arguments.

In order to treat this, we introduce a new notation. '{⟨! arg2⟩, /adv:= {..1..}}' means that the event/object described by this whole description block minus 'adv:={..1..}' corresponds to the arg2 of the description block immediately above, and '/adv:={..1..}' is converted into a predicate at the logical level though it is not the head in Japanese description. By this notation, we can raise non-head elements in one-language descriptions into predicates in logical formulae. Note that our treatment of 'nantoka' is essentially the same as the treatment of 'graag' in the MiMo2 formalism (Noord 1990) so that it has the same defect. That is, it cannot cope with cases where more than two words which require 'raising' like 'nantokea' occur at the same levels (We can avoid this difficulty by introducing different priority levels but this solution seems ad hoc).

5. Definition of Sort Hierarchies

We distinguished in Section 3 two types of logical implications, S-type and C-type. We showed that simple formulae of these two types are expressed in terms of sort hierarchies (S-type) and sortal restrictions on arguments (C-type), respectively. In this section, we introduce notations for more general forms of logical implications.

We will express simple S-type implications by the following notation.

(20) (→ (S)
SUB:[TEACHER:SENSEI]
SUP:[HUMAN:NINGEN])

Because all sort-predicates are one-place predicates (i.e. predicates with arity one), we do not indicate in this notation their argument explicitly by using variables.

Sort-subsort relationships can also be defined among event-sorts such as (21) and (22), which define the sort hierarchy in(23).

(21) (→ (S)
SUB:[WEAR:HAKU]
SUP:[WEAR:MI-NI-TSUKERU])

(22) (→ S)
SUB:[WEAR:KABURU]
SUP:[WEAR:MI-NI-TSUKERU])

(23) [WEAR:MINITSUKERU]

[WEAR:HAKU] [WEAR:KABURU]

The English verb *wear* is a well-known example of a translationally ambiguous word, i.e. when it is translated into Japanese, it can be translated into several different verbs including *haku* ('wear shoes'), *kaburu* ('wear a hat'), *kakeru* ('wear spectacles'), *kiru* ('wear clothes'), etc., depending on what is worn. However, we also have a complex expression *minitsukeru* in Japanese which preserves almost the same vagueness as *wear* has.

Note that though *minitsukeru* is a complex expression as (24) shows, it is mapped into a single predicate because it corresponds to a single lexical item *wear* in English.

(24)	nani-wo	mi-ni	tsukeru.
	[n:something]	[n:body object]	[v:put on location]

The prediacte [WEAR:MI-NI-TSUKERU] can be defined in a way similar to [PAINT:PENKI-WO-NURU].

```
(25) (Def-Pred
     [WEAR:MI-NI-TSUKERU]
     {arg1 := [NINGEN],
      arg2 :=,
      eng  := {head := {e-lex := wear},
               agt := <! arg1>,
               obj := <! arg2>},
      jpn  := {head := {j-lex := tsukeru},
               agt := <! arg1>,
               obj := <! arg2>,
               loc := {head := {j-lex := mi}}}})
```

On the other hand, the definitions of [WEAR:HAKU] and [WEAR: KABURU] have more limited sortal restrictions on the argument-2 as follows.

```
(26) (Def-Pred
     [WEAR:HAKU]
     {arg1 := [NINGEN],
      arg2 := [SHOES:KUTSU],
      eng  := {head := {e-lex := wear},
               agt := <! arg1>,
               obj := <! arg2>},
      jpn := {head := {j-lex := haku},
               agt := <! arg1>,
               obj := <! arg2>}})
```

Though the expression *minitsukeru* can preserve the same vagueness as *wear*, it is desirable in translation to use more specific, less vague target lexical items if such lexical items (like *kaburu*, *haku*, etc) are available.

As Grice pointed out, there is a general principle in human communication that one should use more accurate linguistic forms than less accurate ones if they require the same linguistic effort. If one violates this principle, hearers/readers receive extra information other than their literal meanings (implicatures). This is also the case in translation. If one translates *wear* in *wear shoes* as *minitsukeru* instead of *haku*, the translation sounds less natural and even worse, hearers/readers may infer wrong implicatures (in this case, hearers may infer that the shoes are put on parts of the body other than the feet).

As discussed in the following section, the lexical transfer in our framework is performed first by identifying in the sortal hierarchy the vaguest possible predicates into which a source word can be converted at the logical level and then descending in the hierarchy to find expressions/words with more specific meanings. The process of descending in the hierarchy resembles the process of navigating in discrimination nets in the Schankian approach, but our framework does not decompose meanings of source words into semantic primitives such as M-TRANS, P-TRANS, etc. That is, our framework starts the navigation process from the middle of the hierarchy (not from the top) by using the surface source words as cues.

In the above example, the system first finds the sort [WEAR:MI-NI-TSUKERU] as the vaguest possible predicate into which *wear* can be converted, and then descends the sort-hierarchy like (23) to find more specific words/expressions. In order to facilitate this process, we specify what conditions should be satisfied for descending in the hierarchy through an S-type implication.

Actually, an S-type implication with conditions is a mixed type of S-type and C-type implication such as

(27) (→ [WEAR:HAKU](e)
[WEAR:MI-NI-TSUKERU](e)
& ARG2(e, χ) & [SHOES:KUTSU](χ)]).

This formula says that '[WEAR:HAKU] is a subsort of [WEAR:MI-NI-TSUKERU]' (S-type) and that '[WEAR:HAKU] has a sortal restriction [SHOES:KUTSU] on the argument-2' (C-type). Moreover, in order to descend in the hierarchy, the additional conditions '((ARG2(e,χ) & [SHOES:KUTSU](χ))' in the case of (27)) should be not only necessary conditions but also sufficient conditions for descending. That is, the logical relationship of the two formulae shoud be equivalence but not implication, such as

(28) (↔ [WEAR:HAKU](e)
[WEAR:MI-NI-TSUKERU](e)
& ARG2(e,χ) & [SHOES:KUTSU](χ)),

where ↔ means 'equivalence'.

Supposing that (28) holds, then we express the relationship of the two sorts [WEAR:HAKU] and [WEAR:MI-NI-TSUKERU] in our framework as follows.

(29) (→ [S]
SUB:[WEAR:HAKU]
SUP:[WEAR:MI-NI-TSUKERU]
CON:ARG2(*self*, χ) & [SHOES:KUTSU] (χ)).

The schema (29) expresses that

(i) [WEAR:HAKU] is a subsort of [WEAR:MI-NI-TSUKERU],
(ii) if a event-*self*-belongs to the sort [WEAR:MI-NI-TSUKERU] and if the argument-2 of the event belongs to the sort [SHOES:KUTSU], then the event also belongs to [WEAR:HAKU].

If we use '(S)' instead of '[S]', it shows the relationship is implication but not equivalence, which means (ii) does not hold. (29) is graphically represented as follows.

```
(30) [WEAR:MI-NI-TSUKERU]
               |
     arg2 := [SHOES:KUTSU]
               |
          [WEAR:HAKU]
```

All the event-sorts related with *wear* have the same arity (two), and we assume that the object χ in '[WEAR:HAKU](e) & ARG1 (e, χ)' and the object χ in '[WEAR:MI-NI-TSUKERU](e) & ARG1(e, χ)' denote the same objects. But this continuity of argument structures through sorts is not necessarily guaranteed. A sort can have multiple supersorts and so the continuity of argument structures from different supersorts may conflict with each other.

Furthermore, it is sometimes the case that the arities of events change between a sort and its subsorts. For example, suppose that we have two event sorts [APPLY:NURU] (this event-sort corresponds to the usage of *apply* in *apply glue/paint to ...*) and [PAINT:PENKI-WO-NURU]], and that we define the latter as a subsort of the former. Then, one of the arguments in the supersort [APPLY:NURU] is already given as (15). The definition of [APPLY:NURU] is give as follows.

```
(31) (Def-Pred
     [APPLY:NURU]
     {arg1 :=,
      arg2 := [PAINT:PENKI] v [GLUE:NORI],
      arg3 :=,
      eng  := {head := {e-lex := apply-to},
                 agt := ⟨! arg1⟩,
                 obj := ⟨! arg2⟩,
                 loc := ⟨! arg3⟩},
      jpn  := {head := {j-lex := nuru},
                 agt := ⟨! arg1⟩,
                 obj := ⟨! arg2⟩,
                 loc := ⟨! arg3⟩}})
```

The sort relationship between (15) and (25) is defined as follows.

```
(32) (→[S, (⟨*.ARG2⟩, ⟨ARG2. ARG3⟩)]
     SUB:[PAINT:PENKI-WO-NURU]
     SUP:[APPLY:NURU]
     CON:ARG2(self, χ) & [PAINT:PENKI](χ))
```

'⟨*.ARG2⟩' and '⟨ARG2. ARG3⟩' in this notation mean that the argument2 in the supersort disappears in the subsort and that the argument3 in the supersort is mapped to the argument2 in the subsort. 'ARG*i*' in the CON-part is taken as referring to the argument structures of the supersort for descending from it to the subsorts. Unspecified arguments remain unchanged between the sorts.

6. Inference and Paraphrasing in Translation

Inferences in the discussions so far are all concerned with sort hierarchies. Inferences based on sorts can be implemented efficiently but the forms of inferences are strictly limited. There are several cases (actually a lot of cases, if we want to develop a really high quality MT system) which require more general kinds of inferences.

For example, though we claimed in the above section that English verb *wear* can be translated into appropriate Japanese verbs by using sort restrictions on argument-2 (what is worn) and this can be done by simple forms of inference based on sorts, it is not the case. The conditions for selecting Japanese verbs are more complicated. The conditions are actually not the sorts of objects to be worn, but the place (parts of body) on which the objects are put. The verb *kaburu* is used when the object is put on the head or face, whereas *haku* is used when the object is put on the feet or legs. Because a hat (*booshi*) is usually put on the head and shoes (*kutsu*) on the feet, the verbs *kaburu* and *haku* seem to be used for the sorts [HAT:BOOSHI] and [SHOES:KUTSU], respectively.

However, in a certain situation like a bank hold-up, stockings may be put over the head and face as a disguise. In this case, *kaburu* (which is usually used for hats) should be used instead of *haku* (which is usually used for shoes and stockings). That is, the constraints of the sort [WEAR:HAKU] should be expressed as

$$(33)\ (\rightarrow C\ [\mathrm{WEAR{:}HAKU}](e)\ \ \mathrm{LOC}(e,\chi)\ \&\ ([\mathrm{FEET{:}ASHI}](\chi) \vee [\mathrm{LEG{:}ASHI}](\chi))).$$

Furthermore, the system has to have common sense knowledge something like '*wear shoes* is usually on the feet' and '*wear a hat* is usually on the head', etc. How to use such common sense knowledge (default knowledge) in computer systems and how to make inferences is one of the central issues in AI and it is beyond the scope of this paper. However, we can point out here that, by treating bilingual signs as logical predicates, we can naturally link the translation processes such as disambiguation of translationally ambiguous words, words-to-phrase correspondences, etc. with knowledge-based inference processes.

In translation between such remote language pairs as Japanese and English, we often have to change structures of source sentences completely, which seems to change not only the STRUCTURES but also the MEANING of sentences. Such structural changes cannot be treated only by bilingual signs, but to treat such global structural changes not by linguistic based structure manipulations but by logical inferences.

A well-known example of this type of structural changes is:

(34a)

English :	The typhoon	destroyed	many houses.
	[agt?]		[obj]

(34b)

Japanese:	Taifuu	niyotte	ooku-no	ie-ga	koware	-ta.
	[n: typhoon]	[pp: cause]	[adj: many]	[n:house subject]	[v: collapse]	[tense: past]

As in the case of *like* and *plaire*, we can think that the two structures are the same at the thematic role level by claiming that even in English the subject *typhoon* plays the role of [cause]-case and that in Japanese the thematic role of the syntactic subject *ie* ('house') is [obj]-case. This line of thinking, followed by the Generative Semanticists, leads to the lexical decomposition of *destroy* by combining CAUSE(an abstract predicate) and *collapse*. This attitude is close to the position in MT which represents meanings of sentences by using abstract semantic primitives like Schank's Conceptual Dependency.

However, as already pointed out by many researchers, the paraphrase of (34a) by (35a) looses information which the original sentence conveys (directness of the action). Moreover, even in Japanese, the direct translation (35b) is perfectly grammatical, though it is less natural than the translation of (34b) in most contexts.

(35a) The typhoon caused many houses to collapse.

(35b)

Taifuu-ga	ooku-no	ie-wo	hakaishi
[n: typhoon: agt?]	[abj:many]	[ni:house obj]	[v:destroy]

This means that we need not directly connect the English sentence (34a) directly with (34b). In our framework, the relationship between rather 'literal' but less natural translation like (35b) and 'natural' (as target expressions) translations like (34b) by the relation of logical implication. That is, the relationship between *destroy* and its lexical decomposition which leads to a 'natural' translation is represented as logical implication, as follows.

(36) (→ (S)
[DESTROY:HAKAISURU]
[$(\lambda\, \nu)(\lambda\, w)$
(CAUSE(ν, COLLAPSE:KOWARERU](w)))])

The supersort in this definition is a tentatively defined sort and the tentatively defined defintion shows that this sort requires two arguments in its *internal structure. The arguments of this tentative sort can also be referred* by the predicate ARG1, ARG2, etc. The notation is a little tricky in the sense that the tentative predicate '[$(\lambda\, \nu)(\lambda\, w)$ (CAUSE(ν,[COLLAPSE: KOWARERU](w)))]' itself is a one-place predicate (because this is a predicate for expressing a sort), and the lambda-notation shows its internal argument structure — the variable ν is the argument-1 and the variable w is the argument-2.

Note that '(S)' shows this relation is uni-directional and the transition from [DESTROY:HAKAISURU] to the supersort will loose some information, though the event can still be described.

The system first identifes the sort [DESTROY:HAKAISURU] to be an appropriate sort for the event which the English sentence (34a) expresses. At this stage, it can generate the translation (35b). Then, the system infers by (36) that the same event can be described as another sort, that is, [$(\lambda\, \nu)$...].

In order to materialize the idea that global structural changes in translation be explained by logical inferences, we have to clarify what sorts of inferences are appropriate, what kinds of primitive predicates like CAUSE are necessary, how to link these primitive predicates with linguistic expressions in individual languages, etc.

However, our impression is that the research in MT so far has put too much emphasis on structural issues of translation and has ignored the issues concerned with 'meanings'. As a result, we have tried to bridge two linguistic structures simply by structural operations, even if the two structures differ from each other not only in structures but also in 'contents' (or meanings). The idea of bilingual signs as predicates seems a good start for rectifying this excessive tendency.

7. Sketch of the Transfer Phase

Though we are now at the stage of designing a descriptive framework for bilingual signs, we will give brief sketches in this and following sections of how we will be able to use the descriptions of bilingual signs in actual system.

As we assumed, the monolingual analysis phase generates monolingual descriptions, which are basically thematic role structures of input sentences and on which the transfer phase works. The transfer phase is virtually divided into three subphases as follows, though they can be carried out simultaneously.

(a) Transforming from thematic role structures of source sentences into schema of logical formulae (like (4)).
(b) Determining logical formulae by descending/ascending sort hierarchies: during this phase, inferences based on knowledge are made, and questions are asked to users, if necessary.
(c) Transforming from logical formulae to thematic role structures in the target.

All of these three steps are performed by referring to the definitions of bilingual signs. All the definition of bilingual signs contain the specifications of which parts of linguistic expressions are transformed into the bilingual signs (predicates) and which parts go to their arguments, and vice versa.

We can index each bilingual sign by the surface word whose 'meanings' are expressed by the sign. 'A bilingual sign indexed by a word' means the definition of the bilingual sign can be retrieved by the word. Roughly speaking, a word indexing a bilingual sign is either the word which appears as head in the linguistic form definitions ('eng:=', 'jpn:=', etc.) or the word which is the value in a feature marked by '/' (like *nantoka* in the example [MANAGE:NANTOKA] – See (19) in Section 4).

On the other hand, for each word, we have a set of bilingual signs which are indexed by the word. For *wear*, for example, we have a set of predicates like {[WEAR:MI-NI-TSUKUERU], [WEAR:HAKU], [WEAR:KABRU], ...}. This set is the extension of the predicate schema [WEAR:?]. Because sorts are partially ordered, we can define a set of the vaguest sorts for the word, the vaguest in the sense that none of its supersorts are contained in the extension of the predicate schema.

Step (a) in the above is a rather straightforward process which can be recursively performed through thematic structures. At each recursion level, the system

(i) identifies the (semantic) head of the level,

(ii) retrieves the vaguest possible bilingual signs for the head word, and

(iii) transforms the local structures governed by the head word according to the definition of the bilingual signs retrieved at (ii).

If the level contains words which appear under '/'-features like *nantoka* in the definition of [MANAGE:NANTOKA], the definition of the corresponding 'bi-signs' are first retrieved for checking whether the local structure matches the description. If it matches, the system transforms the local structure accordingly, and then does the normal loop described above.

Because a predicate schema of a word may have several vaguest sorts, step (a) produces several formulae which step (b) tries to transform into more appropriate formulae. The processes of descending in sort hierarchies (disambiguation processes necessary for translation) should be performed for different predicate schemata simultaneously (for verbs and nouns which are related to each other and have mutual sortal restrictions). Though the discussion so far has emphasized the process of descending hierarchies, ascending the hierarchies is also required, because the system has to instantiate all the predicate schemata contained in formulae and each predicate instance has its own sortal restrictons on others. Therefore, step (b) is a kind of relaxation process (relaxation here means to ascend the hierarchies in order to find less restrictive sorts) which tries to find the most accurate solutions ('accurate' means most specific sorts). During this process, some general inference mechanisms may be invoked to infer necessary information for navigating in hierarchies and if necessary, questions will be posed to human users. We are now designing this navigation procedure for simple cases.

The final step, step (c), is rather straightforward, because each bilingual sign identified by step (c) contains in the definition the information on how to transfrom its arguments to target linguistic forms.

8. Disambiguation of Transfer Ambiguities by Paraphrasing

Argument structures between sorts linked in sort-hierarchies are mutually transformable, and each sort definition includes the relationships between its argument structures and the corresponding linguistic structures of both languages.

Because of this explicitness of mutual relationships among sorts and linguistic forms, we can easily express an event (or object) in diversified ways in both languages, the event which is already recognized as an event belonging to one sort. We can express it by seeing it as an event of different sorts (subsorts or supersorts) of the already recognized sort) and do it in both English and Japanese. This paraphrasing facility is very useful for forming and posing appropriate questions during the transfer phase to monolingual speakers of the source language.

Consider the following situation:

(37a) Input sentence: *The teacher runs X.*

(37b) System's knowledge about sorts:

```
                      [RUN:HASHIRASERU]
[EXECUTE:JIKKOOSURU]          |               [MANAGE:KEIEISURU]
        |              ------------------------        |
    [RUN:JIKKOOSURU]                       [RUN:UN'EISURU]
```

As we have already seen, *run* can be translated into several different verbs in Japanese. Suppose that the sort [RUN:HASHIRASERU] is the least specific sort which *run* can describe. An event of this sort can be directly transformed into Japanese expressions by using *hashiraseru.* However, as in the example of *wear* and *minitsukeru,* the direct translation is sometimes awkward if more specific lexical items are appropriate. For example, [ORGANIZATION:SOSHIKI]-*wo hashiraseru* can be understood but only in a very metaphorical way.

The system tries to descend in the hierarchy and find the condition for descending to the sort [RUN:UN'EISURU], i.e. that the argument-2 belongs to the sort [ORGANIZATION:SOSHIKI]. The link between [RUN:HASHIRASERU] has similar sort restrictions on the argument. One possible way of disambiguation by questions is to verbalize these sort restrictions directly such as

(38) Is X an organization or a name of program?

However, what the system has to do here is to disambiguate different senses of *run* which lead to different translations in Japanese, and the system knows that the two senses of *run*, the sorts [RUN:JIKKOOSURU] and [RUN:UN'EISURU], not only share the same supersort [RUN: HASHIRASERU] but also have their own supersorts which are not shared by the other. To verbalize these unshared supersorts can help the system to make the differences explicit. This leads to the following question.

(39) Does the teacher execute X or does the teacher manage X?

We can also think of a question which combines these two types of questions such as

(40) Does the teacher execute X [a program] or does the teacher manage X [an organization]?

What sorts of paraphrasing are really helpful for making ambiguities obvious to monolingual users requires further empirical studies. Though we use only sort hierarchies which are based on the relation of logical implication, it might be the case that the other relations among sorts like 'synonym', 'antonym', etc. are also useful for forming appropriate questions though their logical properties have not been well studied. Whatever relations will be necessary, our idea of bilingual signs which connect the linguistic forms of two languages and general knowledge about events/objects denoted by them (knowledge about sort hierarchies is the simplest one of this type of knowledge) will play a key role in such paraphrasing processes.

References

Alshawi, H. & van Eijck, J. (1989) 'Logical Forms in the Core Language Engine,' *Proc. of 27th ACL*, Vancouver.

Melby, A. K. (1986) 'Lexical Transfer: Missing Element in Linguistic Theories,' in *Proc. of Coling 86.*

Nagao, M. & J. Tsujii & J. Nakamura(1985) 'The Japanese Government Project for Machine Translation,' *Computational Linguistics*, Vol.11, No.2-3.

Nirenburg, S. & I. Nirenburg(1988) 'Framework for Lexical Selection in Natural Language Generation,' in *Proc. of Coling 88.*

Noord, G. & J. Dorrepaal et al. (1990) 'The MiMo2 Research System,' *Proc. of The 3rd International Conference on Theoretical and Methodological Issues in MT.*

Sadler, V. (1990) 'Working with Analogical Semantics,' Distributed Language Translation, Foris.

Tomita, M. (1986) 'Another Stride towards Knowledge-Based MT,' in *Proc. of Coling 86, in Bonn.*

Tsujii, J. (1986) 'Future Directions of MT,' in *Proc. of Coling* 86, in Bonn

Tsujii, J. & M. Nagao(1988) 'Dialogue Translation vs. Text Translation,' in *Proc. of Coling 88.*

Center for Computational Linguistics
UMIST. P.O. Box 88
Manchester M60 1QD
England