

A TRANSFER MODEL USING A TYPED FEATURE STRUCTURE REWRITING SYSTEM WITH INHERITANCE

Rémi Zajac

ATR Interpreting Telephony Research Laboratories
Sanpeidani Inuidani, Seika-cho, Soraku-gun, Kyoto 619-02, Japan
[zajac%atr-ln.atr.junet@uunet.uu.net]

ABSTRACT

We propose a model for transfer in machine translation which uses a rewriting system for typed feature structures. The grammar definitions describe transfer *relations* which are applied on the input structure (a typed feature structure) by the interpreter to produce all possible transfer pairs. The formalism is based on the semantics of typed feature structures as described in [Ait-Kaci 84].

INTRODUCTION

We propose a new model for transfer in machine translation of dialogues. The goal is twofold: to develop a linguistically-based theory for transfer, and to develop a computer formalism with which we can implement such a theory, and which can be integrated with a unification-based parser. The desired properties of the grammar are (1) to accept as input a feature structure, (2) to produce as output a feature structure, (3) to be reversible, (4) to be as close as possible to current theories and formalisms used for linguistic description. From (1) and (2), we need a *rewriting formalism* where a rule takes a feature structure as input and gives a feature structure as output. From (3), this formalism should be in the class of *unification-based formalisms* such as PROLOG, and there should be no distinction between input and output. From (4), as the theoretical basis of grammar development in ATR is HPSG [Pollard and Sag 1987], we want the formalism to be as close as possible to HPSG.

To meet these requirements, a rewriting system for typed feature structures, based on the semantics of typed feature structures described in [Ait-Kaci 84], has been implemented at ATR by Martin Emele and the author [Emele and Zajac 89].

The type system has a lattice structure, and inheritance is achieved through the rewriting mechanism. Type definitions are applied by the interpreter on the input structure (a typed feature structure) using typed unification in a non-deterministic and monotonic way, until no constraint can be applied. Thus, the result is a set of all possible transfer pairs, compatible with the input and with the constraints expressed by the grammar. Thanks to the properties of the rewriting formalism, the transfer grammar is *reversible*, and can even *generate* all possible pairs for

the grammar, given only the start symbol TRANSLATE.

We give an outline of the model on a very simple example. The type inheritance mechanism is mainly used to classify common properties of the bilingual lexicon (sect. 1), and rewriting is fully exploited to describe the relation between a surface structure produced by a unification-based parser and the abstract structure used for transfer (sect. 2), and to describe the relation between Japanese and English structures (sect. 3). An example is detailed in sect. 4.

1. LEXICAL TRANSFER AS A HIERARCHY OF BILINGUAL LEXICAL DEFINITIONS

The type system is used to describe a hierarchy of concepts, where a sub-concept inherits all of the properties of its super-concepts. The use of type inheritance to describe the hierarchy of lexical types is advocated for example in [Pollard and Sag 1987, chap.8].

We use a type hierarchy to describe properties which are common to *bilingual* classes of the bilingual lexicon. The level of description of the bilingual lexicon is the logico-semantic level: a verb for example has a relational role and links different objects through semantic relations (agent, recipient, space-location,...). Semantic relations in the bilingual lexicon are common to English and Japanese.

Predicates can be classified according to the semantic relations they establish between objects. For example, predicates which have only an agent case are defined as Agent-Verbs, and verbs which also have a recipient role are defined as Agent-Recipient-Verbs, a sub-class of Agent-Verbs. On the leaves of the hierarchy, we find the actual bilingual entries, which describe only idiosyncratic properties, and thus are very simple.

The translation relation defined by TRANSLATE is described in sect. 3. We shall concentrate on the propositional part PROP defined here as a disjunction of types:

PROP = SPEAKER | HEARER | REG-FORM | BOOK |
ASK | SEND | TOUCH | NEGATION ...

The simple hierarchy depicted graphically in Figure 1 is written as follows:

```
VERB = [japanese:JV[relation:JPROP],
        english:EJ[relation:EPROP]].
```

```
AG-VERB = VERB[japanese:[agent:#j-ag],
                english:[agent:#e-ag],
                trans-ag:PROP
                [japanese:#j-ag,
                 english:#e-ag]].
```

This definition can be read: an Agent-Verb *is-a* Verb which *has-properties* agent for Japanese and English. We need to express how the arguments of a relation are translated. This is specified using a *translate-ag* slot with type symbol PROP, which will be used during the rewriting process (see details in sect. 3 and 4). Symbols prefixed with # are tags, which are used to represent co-references («sharing») of structures.

In this definition, we have a one-to-one mapping between the agent argument, and at this level of representation (semantic relations), this simple case arises frequently. However, we must also describe mappings between structures which do not have such simple correspondence, such as idiomatic expressions. In that case, we have to describe the relation between predicate-argument structures in a more complex way, as shown for example in sect.4.

```
AG-REC-V = AG-V
           [japanese:[recipient:#j-recp],
            english:[recipient:#e-recp],
            trans-recp:PROP[japanese:#j-recp,
                             english:#e-recp]].
```

```
AG-REC-OBJ-V = AG-REC-V
              [japanese:[object:#j-obj],
               english:[object:#e-obj],
               trans-obj:PROP[japanese:#j-obj,
                               english:#e-obj]].
```

```
NOUN = [japanese:JN, english:EN].
```

Actual bilingual entries are very simple thanks to the inheritance of types.

```
SEND = AG-REC-OBJ-V[japanese:[reln:OKURU-1],
                    english:[reln:SEND-1]].
```

```
ASK = AG-REC-V[japanese:[reln:OKIKI-1],
               english:[reln:ASK-1]].
```

```
REG-FORM = NOUN
           [japanese:TOUROKUYOUSHI-1,
            english:REGISTRATION-FORM-1].
```

```
B-HEARER = NOUN[japanese:J-HEARER,
                 english:E-HEARER].
```

```
B-SPEAKER = NOUN[japanese:J-SPEAKER,
                  english:E-SPEAKER].
```

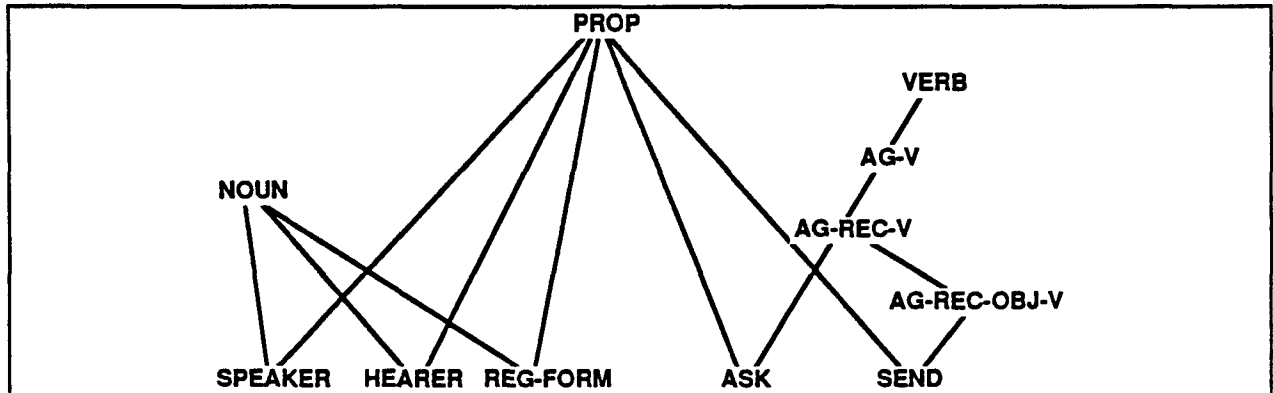


Figure 1: a simple hierarchy of types.

The type system is interpreted using the rewriting mechanism described in [Ait-Kaci 84], which gives an operational semantics for type inheritance: a feature structure which has a type AG-V for example is unified with the definition of this type:

```
[japanese:[agent:#j-ag],
  english:[agent:#e-ag],
  trans-ag:PROP[japanese:#j-ag,
                 english:#e-ag]]
```

and the type symbol AG-V is replaced with the super-type VERB in the result of the unification. If type VERB has a definition, the structure is further rewritten, thus achieving the operational interpretation of inheritance. Disjunctions like PROP create a non-deterministic choice for further rewriting: the symbol PROP is replaced with the disjunction of symbols of the right-hand-side creating alternative paths in the rewriting process. This process of rewriting is applied on every

sub-structure of a structure to be evaluated, until no type symbol can be rewritten.

As the rewriting system does not have any explicit control mechanism for rule application, whenever several rules are applicable all paths are explored, and all solutions are produced in a non deterministic way. This could be a drawback for a practical machine translation system, as only one translation should be produced in the end, and due to the non deterministic behavior of the system, this could also lead to severe efficiency problems. However, the system is primarily intended to be used as a tool for developing a linguistic model, and thus the production of all possible solutions is necessary in order to make a detailed study of ambiguities.

Furthermore, according to the principles of second generation MT systems [Ynvg 57, Vauquois 75, Isabelle and Macklovitch 86], a transfer grammar should be purely contrastive, and should not include specific source or target language knowledge. As a result, the synthesis grammar should implement all

watashi-ni	tourokuyoushi-wo	o-okuri	itadake	masu	ka
<i>I-dative</i>	<i>registration-form-acc</i>	<i>honor-send</i>	<i>can-receive-a-favor</i>	<i>polite</i>	<i>interr</i>

Figure 2: example of a Japanese sentence

The representation has already categorized to a certain extent surface speech acts types. The level of analysis produced by the parser is the level of semantic relations (relation, agent, recipient, object,...). The representation reduced to relation features is:

```
(INTERROGATE-IF (CAN (RECEIVE-FAVOR (OKURU-1
(TOUROKUYOUSHI-1)))) )
```

The level of representation we want for transfer can be basically characterized by (1) an abstract speech act type (request, declaration, question, promise,...), (2) a manner (direct, indirect,...), and (3) the propositional content of the speech act [Kume et al. 88]. A grammar, written in the same formalism, abstracts the meaning of the surface structure to:

```
JACA[speech-act-type: REQUEST,
manner: INDIRECT-ASKING-POSSIBILITY,
speaker: #speaker=J-SPEAKER,
hearer: #hearer=J-HEARER,
s-act: JV[relation: OKURU-1,
agent: #hearer,
recipient: #speaker,
object: TOUROKUYOUSHI-1]]
```

and this is the input for the transfer module.

necessary language specific constraints in order to rule out ungrammatical structures that could be produced after transfer, and make appropriate pragmatic decisions.

2. RELATING SURFACE AND ABSTRACT SPEECH ACTS

A problem in translating dialogues is to translate adequately the speaker's communicative strategy which is marked in the utterance, a problem that does not arise in text machine translation where a structural translation is generally found sufficient [Kume et al. 88]. Indirectness for example cannot be translated directly from the surface structure produced by a syntactic parser and needs to be further analyzed in terms independent of the peculiarities of the language [Kogure et al. 1988]. For example, take the representation produced by the parser for the sentence [Yoshimoto and Kogure 1988]:

3. DEFINING THE TRANSFER RELATION AT THE LOGICO-SEMANTIC LEVEL

Each structure which represents an utterance has (1) an abstract speech act type, (2) a type of manner, and (3) a propositional content. Each sub-structure of the propositional content has (1) a lexical head, (2) a set of syntactic features (such as tense-aspect-modality, determination, gender,...), and may have (3) a set of dependents which are analyzed as case roles (agent, time-location, condition,...).

The manner and abstract speech act categories are universals (or more exactly, common to this language pair for this corpus), and need not be translated: they are simply stated as identical by means of tag identity.

The part which represents the propositional content is language dependant, and the translation relation defined between lexical heads, syntactic features and dependents of the heads is defined indirectly by means of transfer rules. Thus, this approach can be characterized as a mix of pivot and transfer approaches [Tsuji 87, Boitet 88].

<i>speech-act-type</i> REQUEST <i>manner</i> INDIRECT-ASK-POSSIBILITY	direct mapping by tagging	<i>speech-act-type</i> REQUEST <i>manner</i> INDIRECT-ASK-POSSIBILITY
speaker #0=J-SPEAKER hearer #1=J-HEARER s-act relation OKURU-1 agent #1 recipient #0 object TOUROKUYOUSHI-1	indirect mapping by rule application	speaker #2=E-SPEAKER hearer #3=E-HEARER s-act relation SEND-1 agent #3 recipient #2 object REGISTRATION-FORM-1

Figure 3: the translation relation.

The definitions of the transfer grammar can be divided into three groups:

- 1) definitions that state equality of abstract speech act type and manner (the language independent parts),
- 2) lexical definitions that relate predicate-argument structures,
- 3) definitions that relate syntactic features (not yet included in our grammar).

Starting from the abstract speech act description, we need only one definition for specifying the direct mapping of Abstract Speech Acts by tagging, which also introduces the type symbol PROP that will trigger the rewriting process for the transfer grammar:

```

TRANSLATE =
[japanese:JASA
  [speech-act-type: #sat,
   manner: #manner,
   speaker: #j-spk,
   hearer: #j-hrr,
   s-act: #j-act=JPROP]],
english:EASA
  [speech-act-type: #sat,
   manner: #manner,
   speaker: #e-spk,
   hearer: #e-hrr,
   s-act: #e-act=EPROP]],
trans-act:PROP[japanese:#j-act,
               english:#e-act]],
trans-spk:PROP[japanese:#j-spk,
               english:#e-spk]],
trans-hrr:PROP[japanese:#j-hrr,
               english:#e-hrr]].

```

In this simple example, the definition of the symbol PROP contains the full bilingual dictionary. Unifying a structure with PROP means that a structure is unified with a very large disjunction of definitions. There are several possible ways to overcome this problem. One can use the hierarchical type system to restrict the set of candidates to a small sub-set of definitions and instead of using PROP, use the most adequate specific symbol for translating an argument: such a symbol can be viewed as the initial symbol of a sub-grammar which describes the transfer relation on a

sub-class of lexemes. For example, one can write directly SPEAKER instead of PROP in the trans-spk slot of the above definition. Another possibility for a mono-directional system is to access the bilingual lexicon using the Japanese entry during parsing. This means that the dictionaries of the system would have to be organized as a single integrated bilingual lexical database.

4. A STEP BY STEP EXAMPLE

We give in this section a trace of a simple example for the sentence in Figure 4. For translating, we need to add to the definition of PROP, the following bilingual lexical definitions:

```

BOOK = NOUN[japanese:HON-1, english:BOOK-1].
HAND = NOUN[japanese:TE-1, english:HAND-1].

TOUCH = VERB
[japanese:[relation: FURERU-1,
          object: TE-1,
          spatial-destination: #0],
 english: [relation: TOUCH-1,
          object: #1],
 trans0:PROP[japanese: #0, english:#1]].

```

hon-ni	te-wo	fure-naide	kudasai
book-obj2	hand-obj1	touch-neg	please

Figure 4: don't touch the books!

A lexical definition introduces the PROP symbol for the arguments of a predicate, and the translation relation is defined recursively between argument sub-structures. There could be one-to-one mapping between two sub-structures, but as in the example of TOUCH, the relation is in general not purely compositional, and not one-to-one, and argument description can be as refined as necessary. Here, the object TE-1 («hand») is a part of the meaning of «touch» in this kind of construction, and the semantic relation that links the predicate and the object being touched is a spatial destination in Japanese (perceived as a goal or a target) and an object in English.

INPUT : a structure representing a deep analysis of the sentence in Figure 4. The initial symbol that will be rewritten is TRANSLATE (symbols to be rewritten are in bold face).

TRANSLATE

```
[japanese: JASA
 [speech-act-type: #sat=REQUEST,
  manner: #man=DIRECT,
  speaker: #j-sp=J-SPEAKER,
  hearer: #j-hr=J-HEARER,
  s-act:#j-act=
    [relation: NEGATE
     object:
      [relation: FURERU-1,
       object: TE-1,
       spatial-destination: HON-1]]]
```

STEP 1: rewrite TRANSLATE which adds to the input structure the English ASA and new PROP symbols in the translate-act, trans-speaker and trans-hearer slots.

```
[japanese: JASA
 [speech-act-type: #sat=REQUEST,
  manner: #man=DIRECT,
  speaker: #j-sp=J-SPEAKER,
  hearer: #j-hr=J-HEARER,
  s-act:#j-act=J-PROP
    [relation: NEGATE
     object:
      [relation: FURERU-1,
       object: TE-1,
       spatial-destination: HON-1]]]
english: EASA
 [speech-act-type: #sat,
  manner: #man,
  speaker: #e-speaker,
  hearer: #e-hearer,
  s-act:#e-act=EPROP],
trans-act: PROP
  [japanese: #j-act,
   english: #e-act], ...]
```

STEP 2 and 3 : the new PROP symbols are rewritten as disjunctions. For the s-act slot, the unification with NEGATION is successful. It adds a new PROP symbol which is in turn rewritten and this time the unification with TOUCH succeeds: it adds the English object and a new translate slot for BOOK.

```
[japanese: JASA
 [speech-act-type: #sat=REQUEST,
  manner: #man=DIRECT,
  speaker: #j-sp=J-SPEAKER,
  hearer: #j-hr=J-HEARER,
  s-act:#j-act=J-PROP
    [relation:#j-neg=J-NEG
     object:#-obj1
     [relation: FURERU-1,
      object:#j-obj2=TE-1,
      spatial-destination:#sd=HON-1]]]
english: EASA
 [speech-act-type: #sat=REQUEST,
  manner: #man=DIRECT,
  speaker: #j-sp=E-SPEAKER,
  hearer: #j-hr=E-HEARER,
  s-act:#e-act=EV
    [relation:#e-neg=E-NEG,
     object:#e-obj=
      [relation: TOUCH-1,
       object:#e-obj2]]],
trans-act: ...,
trans-obj: [japanese: #j-obj1,
            english: #e-obj,
            trans0: PROP [japanese: #sd,
                          english: #e-obj2]]]
```

STEP 4 : the new PROP symbol is in turn rewritten as BOOK which finally translates the last argument. The final structure produced by the interpreter is:

```
[japanese: JASA
 [speech-act-type: #sat=REQUEST,
  manner: #man=DIRECT,
  speaker: J-SPEAKER,
  hearer: J-HEARER,
  s-act: J-PROP
    [relation: J-NEG
     object:
      [relation: FURERU-1,
       object: TE-1,
       spatial-destination: HON-1]]],
english: EASA
 [speech-act-type: #sat,
  manner: #man,
  speaker: E-SPEAKER,
  hearer: E-HEARER,
  s-act: E-PROP
    [relation: E-NEG,
     object:
      [relation: TOUCH-1,
       object: BOOK-1]]],
...]
```

CONCLUSION

The rewriting formalism has been implemented in LISP by Martin Emele and the author at ATR in order to develop transfer and generation models of dialogues for a machine translation prototype [Emele and Zajac 89]. The two main characteristics of the formalism are (1) type inheritance which provides a clean way of defining classes and sub-classes of objects, (2) the rewriting mechanism based on typed unification of feature structures which provide a powerful and semantically clear means of specifying (and computing) *relations* between classes of objects. This later behavior is somehow similar to the PROLOG mechanism, and grammars can be written to be reversible, which is the case for our transfer grammar. We hope this feature will be useful in the future development of the grammar, allowing for a precise contrastive analysis of Japanese and English.

At present, the transfer grammar is in a very early stage of development but nevertheless, capable of translating a few elementary sentences. It covers basic sentence patterns; compound noun phrases and coordination of noun phrases; verb phrases including auxiliaries, modals and adverbs; sentence adverbials; conditionals.

The transfer module and the generation module [Emele 89] use the same formalism and integration is thus simple to achieve. As for efficiency considerations, the transfer and generation of the sentence in Figure 2 takes approximately 5 seconds on a Symbolics with our current implementation. However, this figure is not very meaningful because our dictionaries and grammars are still very small, and the implementation of the interpreter itself is still evolving.

Full integration with the analysis module (a unification-based parser which produces a set of feature structures) remains to be worked out, but should not cause major problems. In this respect, the closest related works are a transfer model proposed by [Isabelle and Macklovitch 86] and a model in the LFG framework proposed by [Kudo and Nomura 86] (see also [Beaven and Whitelock 88]).

There are two major topics for further research:

- 1) the extension of the formalism to include full logical expressions, as described for example in [Smolka 88], and some kind of control mechanism in order to treat default values and prune some solutions (when an idiomatic expression is found for example);
- (2) the development of a transfer grammar for a larger language fragment, using outputs of the parser already available described in [Yoshimoto and Kogure 1988].

REFERENCES

- Hassan AIT-KACI. 1984. *A Lattice Theoretic Approach to Computation Based on a Calculus of Partially Ordered Type Structures*. Ph.D. Thesis, University of Pennsylvania.
- John L. BEAVEN and Pete WHITELOCK. 1988. Machine Translation Using Isomorphic UCGs. *Proceedings of COLING-88*, Budapest.
- Christian BOITET. 1988. Pros and Cons of the Pivot and Transfer Approaches in Multilingual Machine Translation. *Proc. of the Intl. Conf. on New Directions in Machine Translation*, BSO, Budapest.
- Martin EMELE. 1989. A Typed Feature Structure Unification-based Approach to Generation. *Proceedings of the WGNLC of the IECE*, Oita University, Japan.
- Martin EMELE and Rémi ZAJAC. 1989. *RETIF: a Rewriting System for Typed Feature Structures*. ATR Technical Report TR-I-0071.
- Pierre ISABELLE and Eliot MACKLOVITCH. 1986. Transfer and MT Modularity. *Proceedings of COLING-86*, Bonn.
- Kiyoshi KOGURE, Kei YOSHIMOTO, Hitoshi IIDA, and Teruaki AIZAWA. 1988. *The Intention Translation Method, A New Machine Translation Method for Spoken Dialogues*. Submitted for IJCAI-89, Detroit.
- Ikuo KUDO and Hirosato NOMURA. 1986. Lexical-Functional Transfer: A Transfer Framework in a Machine Translation System based on LFG. *Proceedings of COLING-86*, Bonn.
- Masako KUME, Gayle K. SATO and Kei YOSHIMOTO. 1988. A Descriptive Framework for Translating Speaker's Meaning. *Proceedings of the 4th Conference of ACL-Europe*, Manchester.
- Carl POLLARD and Ivan A. SAG. 1987. *Information-based Syntax and Semantics*. CSLI, Lecture Notes Number 13, Stanford.
- Gert SMOLKA. 1988. *A Feature Logic with Subsorts*. LILOG-REPORT 33, IBM Deutschland GmbH, Stuttgart.
- Jun-Ichi TSUJII. 1987. What is pivot?, *Proceedings of the 1st MT Summit*, Hakone.
- Bernard VAUQUOIS. 1975. *La traduction automatique à Grenoble*. Document de Linguistique Quantitative 29, Dunod, Paris.
- V.M. YNVEGE. 1957. A Framework for Syntactic Translation. *Mechanical Translation* 4/3, 59-65.
- Kei YOSHIMOTO and Kiyoshi KOGURE. 1988. *Japanese Sentence Analysis by means of Phrase Structure Grammar*. ATR Technical Report TR-I-0049.