INDEPENDENT TRANSFER USING GRAPH UNIFICATION

Lauri Carlson
Maria Vilkuna

Department of General Linguistics
University of Helsinki
Hallituskatu 11
00100 Helsinki, Finland
lcarlson@finuh.bitnet , vilkuna@finuh.bitnet

Abstract

We present a MT system that applies graph unification in transfer from English to Finnish. The work described below is an outgrowth of a multilingual MT project initiated by the IBM in 1987 with the aim of studying multilingual translation using a common English language parser.

The transfer system presented here is independent of the parsing and generation modules. Any source language parser can be used whose output can be expressed in a directed graph form. The transfer system is responsible for generating target language phrase structure. Target language word order and morphology are left to the generation modules.

The transfer system is lexically based. Transfer rules, presented in the form of bilingual graphs, are declarative statements of symmetric transfer relationships between words, phrases or constructions in the two intertranslatable languages.

Transfer is structure driven in that the transfer algorithm traverses the source language graph, nondeterministically trying to apply the relevant transfer rules in the lexicon. Each successful transfer yields a bilingual graph, whose target language half is extracted and subjected to linearization and morphological generation.

The main focus of attention in our project is the development of the lexicon subsystem. The lexicon system consists of separate transfer and monolingual lexicons and a common lexicon of language independent definitions.

Keywords: unification, machine translation, transfer, bilingual lexicon

1. Unification based transfer

Our approach is more transfer oriented than some other unification based approaches to MT (e.g., Beaven and Whitelock 1988). However, we argue, use of graph unification blurs the distinction between transfer and interlingua.

A feature structure representing a phrase will contain information at several levels of linguistic analysis ranging from lexical identity to logical argument structure. Transfer rules can express bilingual correspondences at any level of abstraction as well as across different levels of structure. (Cf. Kaplan & al. 1989.) A transfer rule in our sense can consist of an arbitrary pairing of lexical entries, a complex correspondence across structures (e.g., "change" of grammatical construction including part of speech assignments), or a straightforward identification of arguments in logical form.

When the translation relation is best stated in language independent (semantic) terms, transfer is trivial. Then monolingual lexicons, analysis and generation modules will do most of the work. Thus, to what extent a given rule has the character of a genuine transfer rule will depend on the degree of similarity of the languages under translation in the relevant respect. For instance, languages with similar tense systems can allow a straightforward identification of low level tense distinctions. Low level transfer simplifies the tasks of analysis and generation and allows tighter control of the translation relation. In particular, transfer idioms (multiword equivalences) can be stated directly without a detour through more abstract representations. In this sense, unification based transfer fills out the space separating interlingua and transfer.

2. Parsing

Unlike approaches such as Kaplan & al (1989), which produce bilingual descriptions in the course of parsing source language text, transfer in our system has a completed parse as a starting point. Currently, this parse is produced by a general-purpose parser, PEG of IBM T.J.Watson Research Center (Jensen 1986), which is not unification-based. However, its output is close enough to a directed graph to allow conversion into the form required by the transfer system using a simple conversion interface.

It appears to us that this decoupling of parsing from transfer is a safe move. Knowledge of the target language is not likely to influence parsing of the source language in any significant fashion[1].

3. The transfer system

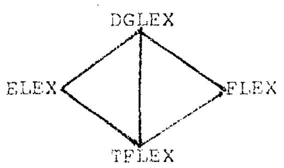
Our transfer system consists of two modules. A declarative module defines translation correspondences of individual phrases, structures and features. The information is given in bilingual (or multilingual) transfer dictionaries.

An algorithmic module actually builds the correspondence structure out of the source language f-structure and the transfer dictionaries. This component ensures that all necessary alternatives are considered and the relevant information is incorporated into a correct location in the correspondence structure.

We discuss these two modules in turn.

3.1. The transfer lexicon

A leading idea of the lexicon system is the separation of four different lexicons as follows:



DGLEX is a lexicon of general linguistic definitions of terms. There are two monolingual lexicons, ELEX and FLEX, and a bilingual transfer lexicon, TFLEX. The monolingual lexicons depend on DGLEX, and TFLEX can refer to the other three. No further dependencies are allowed. This increases the independence between the component lexicons and makes them reusable for multilingual translation.

The descriptions in both monolingual lexicons are kept independent of one another and linguistically motivated. Complex and ad hoc statements belong in TFLEX; it cannot be expected that all bilingual intertranslatability relations should follow linguistic generalizations. Correspondingly, we may distinguish two kinds of multi-word expressions. Language-internal idioms (e.g., *keep tabs* in English) are given in the monolingual lexicons, whereas the other type, which might be called "transfer idioms", are referred to at the level of transfer entries only (e.g., *have access to*, which translates into one Finnish verb).

3.2. The specification language

The linguistic description language has two levels, an internal representation in terms of attribute value graphs, and a definition language consisting of templates abbreviating such graphs. As examples of the latter, consider the simple entries below.

```
(1) (discuss v simpleobj-e)
(2) (keskustella v simpleobl-ela)
(3) (discuss  (e (@ e::discuss))
              (f (@ f::keskustella))
              tra)
```

```
[E:[LEX:BE
    CAT:VERB
    SUBJ:#3[E:[LEX:IT
               CAT:PRON
               SEM:#2[ANIM:F
                      HUM:F]]
            F:[LEX:SE
               CAT:PRON
               CASE:ELA
               SEM:#2]]
    VCOMP:#4[E:[LEX:DISCUSS
                CAT:VERB
                SUBJ:#3
                PRED:[ARG1:
                      #5[E:[LEX:*NONE*]
                         F:[LEX:*NONE*
                            SEM:[HUM:T]]]
                      ARG2:#3
                      ARG3:*NONE*]
                VFORM:PASTPART
                VOICE:PASS]
             F:#10[LEX:KESKUSTELLA
                   CAT:VERB
                   THEMA:#3
                   SUBJ:#5
                   OBL:#3
                   PRED:[ARG1:#5
                         ARG2:#3
                         ARG3:*NONE*]
                   VFORM:FINITE
                   VOICE:PASS]]
    PRED:[ARG1:#4
          ARG2:*NONE*
          ARG3:*NONE*]
    VFORM:FINITE
    VOICE:PASS]
 F:#10]
```

Fig. 1: Simplified TFS of "it was discussed" (next page)

The entries are from ELEX, FLEX, and TFLEX, respectively, and together they specify the transfer relation between English *discuss* and its Finnish equivalent *keskustella***.** (The transfer entry is shown expanded into graph form in fig. 4.)

The graph formalism we use is a standard attribute value unification formalism except for the use of cyclic graphs. The graph specification language extends the template language used in D-PATR in the following respects:

- Compile-time disjunction is included
- Parametric templates are included

3.3. Transfer feature structures (TFS)

The transfer relation between source and target language feature structures could be represented in different ways. Separate feature structures could be set up for the source language and the target language, and an explicit transfer relation between these two structures could be defined (Kaplan & al. 1989). In our system, there is only one larger transfer feature structure (TFS) which includes both feature structures and specifies the explicit transfer relation for intertranslatable phrases of source and target languages.

The TFS contains extra levels of attributes for the source and target language. Intertranslatable phrases form subdescriptions which have two attributes, one for each language. The values of these attributes are always trans-

```
[F:#10[LEX:KESKUSTELLA
       CAT:VERB
       THEMA:#3[F:[LEX:SE
                   CAT:PRON
                   CASE:ELA
                   SEM:#2[ANIM:F
                          HUM:F]]]
       SUBJ:#5[F:[LEX:*NONE*
                  SEM:[HUM:T]]]
       OBL:#3
       PRED:[ARG1:#5
             ARG2:#3
             ARG3:*NONE*]
       VFORM:FINITE
       VOICE:PASS]]
```

Fig. 2: Simplified Finnish FS of 'It was discussed'

lations of each other, and they may share values of common features and especially component phrases which, in turn, are translations of each other.

An example of a translation relation expressed in one feature structure is given in fig. 1. This structure contains the feature descriptions of both the English and Finnish sentences and coreferential links that bind the corresponding units together.

Monolingual feature representations can be read off the bilingual one by omitting all attribute-value pairs where the attribute is the name of the other language. The Finnish language subgraph of the previous example is given in fig. 2.

```
[E:[TENSE:#1]
 F:[TENSE:#1]]
```

Fig. 3: Simple tense transfer rule

```
[E:[LEX:DISCUSS
    CAT:VERB
    SUBJ:#2[E:[DUMMY:F]]
    OBJ:#3[F:[CASE:ELA]]
    PRED:[ARG1:#2
          ARG2:#3
          ARG3:*NONE*]
 F:[LEX:KESKUSTELLA
   CAT:VERB
   SUBJ:#2
   OBL:#3
   PRED:[ARG1:#2
         ARG2:#3
         ARG3:*NONE*]]]
```

Fig. 4: Partial transfer rule for "discuss"

```
[E:[LEX:BE
    SUBJ:#2
    VCOMP:#5[E:[SUBJ:#2
                BY-PASS:F
                VFORM:PASTPART
                VOICE:PASS]
             F:#1[THEMA:#2
                  SUBJ:[F:[LEX:*NONE*
                           SEM:[HUM:T]]]
                  VOICE:PASS
                  NOMOBJ:T]]
```

Fig. 5: Simplified transfer rule for agentless passive

3.4. Transfer rules

A transfer rule in this approach is formally just another transfer feature structure, similar to the bilingual structure. It is a partial specification of an acceptable intertranslatability relation. The rule is applied to a TFS by unifying it with a specified node in the TFS. The transfer process consists simply of adding of further information into a partially described instance of the transfer relation. There is no formal distinction between lexical and grammatical transfer rules. Examples of different types of rule are given in figures 3-5.

Some aspects of our linguistic description will be briefly described. In monolingual lexicons, shifts in grammatical function like the English active and passive are described as different linkings of arguments to grammatical functions, in this case, the subject and the object function.

In transfer of complement-taking elements, we can then for the most part rely on the simple rule "equate arguments", which results in correct bilingual correspondences given the language-particular linkings. For example, the verb *discuss* (fig. 4) takes as its second argument a direct object in English but an oblique complement in Finnish, but this language-particular information need not be recapitulated in the transfer entry.

There are also translation equivalents whose arguments do not match, and these receive slightly more complex transfer rules where argument equations are expressed separately.

Graph unification descriptions are particularly simple and effective where the relevant structures consist of predicates taking a restricted number of unique argument types, such as subject, object, or sentential complement.

Adjuncts, which may have multiple instantiations for each head, need a different treatment. Each of the adjuncts has a unique modifiend (modif = the modified word),

```
#1[E:[LEX:EXAMPLE
      CAT:NOUN
      ADJT:#2[E:[LEX:ADDITIONAL
                 CAT:ADJ
                 PRED:[ARG1:#1
                       ARG2:*NONE*
                       ARG3:*NONE*]
                 ADJT:
                     [E:[CAT:ADV
                         MODIF:#2]]

                 MODIF:#1]
              F:[LEX:LISA
                 CAT:NOUN
                 ADJT:[F:*NONE*]
                 MODIF:#1
                 NUM:SG]]
       NUM:PL
       PERS:3
   F:[LEX:ESIMERKKI
       CAT:NOUN]]
```

Fig. 6: A cyclic TFS

which it may share with other adjuncts. We allow adjuncts to point back to the modifiend so as to let transfer rules refer to properties of the modifiend. This means that a TFS can be a cyclic graph. This is illustrated in fig. 6.

4. Generation

Since complex aspects of the transfer mapping are handled by the parser and the transfer system, generation in our model remains simple. It involves a recursive sort of the lexical entries of the target language and the generation of morphologically inflected forms from sets of morphological features.

The linearization component uses a set of unification based LP rules operating on information in the final Finnish feature structure. Discourse-related information relevant for linearization is included in the feature structure.

For Finnish subjectless clause types, we use a transfer rule that requires equation of the English subject with the Finnish discourse function THEMA. Depending on clause type, any one of the Finnish arguments may appear as a THEMA (e.g., "about it one-must discuss"; see fig. 7). The linearization rule then places the THEMA before the finite verb, preserving, in effect, the characteristic information structure of the English sentence.

```
[F:[LEX:TAYTYA 'must'
    CAT:VERB
    THEMA:#3[F:[LEX:SE 'it'
                CAT:PRON
                CASE:ELA]]
    VCOMP:#9[F:#10[LEX:KESKUSTELLA
                             'discuss'
                   CAT:VERB
                   THEMA:#3
                   SUBJ:#5[F:[LEX:*NONE*
                              SEM:[HUM:T]]]
                   OBL:#3
                   VFORM:INF1
                   VOICE:ACT]]
    VFORM:FINITE
    VOICE:ACT]]
```

Fig. 7: A Finnish impersonal. Thema percolated from VCOMP

Morphological generation involves production of Finnish inflected word forms from morphological tags obtained from the Finnish feature structrue using Koskenniemi's two-level morphological processor.

5. Conclusion

The choice of unification as a descriptive tool in developing the transfer lexicon system has been productive in our experience. In conclusion, we survey the properties of graph unification that have proved valuable.

- Recursive structure of TFS: No limit to the complexity of an entry. Multiword entries on a par with one word entries.
- Uniformity: Linguistic information at different levels represented in a uniform way. No dichotomy of lexical and structural transfer.
- Unification: Structure changing correspondences can be expressed through coindexing.
- Subsumption: Inheritance of definitions allows making generalisations across entries and lexicons.
- Partial information: No requirement of completeness of linguistc descriptions for transfer to work. Disjunctions eliminated by underspecification. No need to make translation related sense distinctions in monolingual lexicons.
- Monotonicity: Entries remain valid when lexicon is extended and enriched. Enables incremental refinement of individual entries and grammatical correspondences.
- Commutativity and associativity: Entries remain valid when entries or sense definitions are rearranged or regrouped.

Notes

1 Since unification-based transfer is monotonic, the assumption of completeness of input is not essential for us. Nothing in principle rules out incremental transfer during parsing.

Acknowledgements

This research has been supported by IBM Finland. We thank Kimmo Koskenniemi for his insights at the system planning stage and to Krister Lindén for discussions on this paper.

References

Beaven, J.L. - Whitelock, P. 1988: Machine Translation Using Isomorphic UCGs. Proceedings of COLING '88, Budapest.

Jensen, K. 1986: PEG 1986: A Broad-coverage Computational Syntax of English. Technical Report, IBM T.J. Watson Research Center.

Kaplan, R. - Netter, K. - Wedekind, J. - Zaenen, A. 1989: Translation by Structural Correspondendes. Proceedings of the Fourth Conference of the European Chapter of ACL, Manchester.