Structural and lexical transfer from an UNL graph to an equivalent natural language dependency tree

Etienne Blanc, Gilles Sérasset, WangJu Tsai

GETA, CLIPS-IMAG
BP 53, F-38041 Grenoble cedex 09
{etienne.blanc, gilles.serasset, tsai}@imag.fr

Abstract

We describe the transfer of an UNL graph into a equivalent tree, allowing to build UNL deconverters using existing MT systems based on tree processing.

1. Introduction

In the Universal Networking Language, a text is represented as a graph where nodes, bearing "Universal Words" (UWs), are linked by directed arcs bearing semantic "Relations Labels". A particular node, the "entry node", is distinguished in the graph.

The structure of these UNL graphs makes them quite suited to be processed by various linguistic tools. In particular, the Deconversion (from a UNL graph into an equivalent Natural Language text) or the Enconversion (from a Natural Language text into a UNL graph) may be achieved not only using the specially devised Deco and Enco tools, but also using adapted existing classical MT systems. For instance, UNL to Russian, UNL to Chinese, UNL to French deconverters are being developed using transfer MT systems.

Most of the classical MT systems use tree representation and not graph representation. Therefore the first step in the deconversion based on such systems is a graph-to-tree transfer. The aim of this paper is to discuss such a transfer, and to present the method used in the UNL-to-French deconverter.

We will begin by an overall presentation of the UNL-to-French deconvertor based on the ARIANE-G5 generator of MT systems. We will then discuss in more detail the process of graph-to-tree transfer.

2. A UNL-to-French deconverter deriving from a classical transfer system

2.1. Ariane-G5, a generator of MT systems

ARIANE-G5 is a generator of MT systems, that is an integrated environment designed to facilitate the development of MT systems (Boitet, 1997). These MT systems are written by a linguist using specialized languages for linguistic programming. ARIANE is not devoted to a particular linguistic theory. The only strong constraint is that the structure representing the unit of translation (sentence or paragraph) must be a decorated tree.

Fig.1 shows an overview of a classical transfer MT system using the ARIANE environment. The processing is performed through the three classical steps : analysis, transfer and generation.

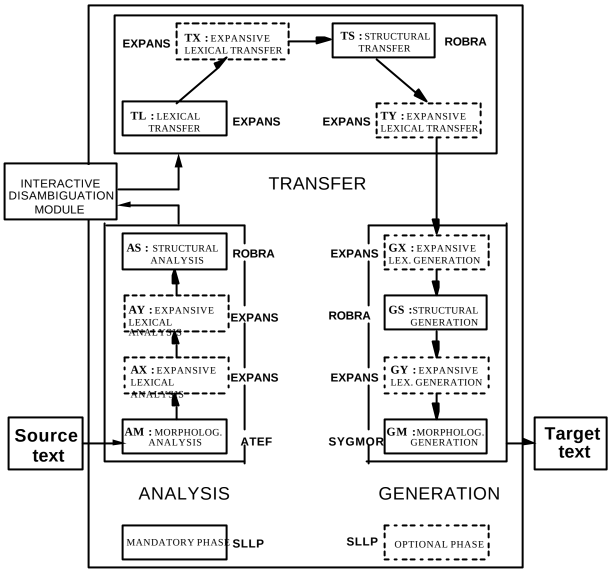


Figure.1 The Ariane-G5 environment as used for generating a transfer MT SYSTEM

2.2. Principle of the French Deconverter

Fig 2 shows an overview of the UNL-to-French deconverter using the ARIANE environment.

The first step is a graph-to-tree transfer, achieving both:

- the graph-to-tree structural transfer necessary for the ulterior Ariane processing
- a lexical "Universal Words" to French words lexical transfer.

The resulting tree is a classical "deep tree" ready for generation.

This first structural and lexical step will be discussed in detail below. The following classic generation step will not be discussed here.

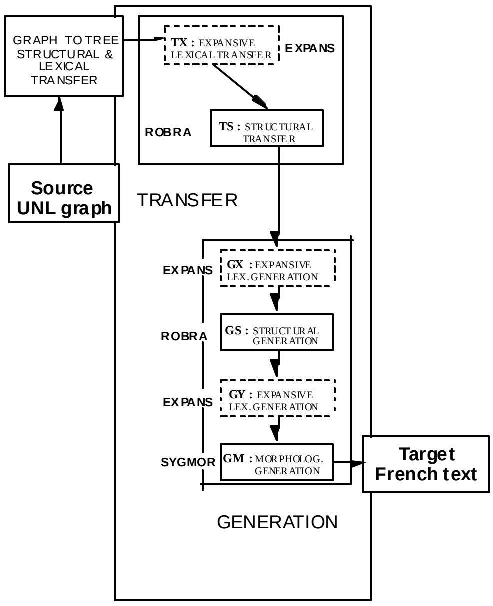


Figure 2 : The Ariane-G5 environment as used for generating a French deconverter.

3. UNL graph to NL tree structural transfer

The aim of the graph-to-tree structural transfer is to supply an output tree displaying all the structural information contained in the input UNL graph.

We will consider the following examples of tructural features encountered in a graph and needing some special coding in a tree are for instance:

- node having several mother nodes
- closed circuit
- hypergraph structure, that is graph containing nodes having themselves a graph structure (subgraphs, or "Compound Universal Words")

But before considering these examples, let's first illustrate the transfer on the simplest case, that is the transfer of a graph having in fact already a tree structure.

3.1. Graph with tree structure

In this simple case, the transfer is straightforward, as illustrated on figure 3.

This figure gives successively, from top to bottom:

- the meaning of the input graph as expressed in English
- the graph itself
- a sketch of its structure
- the structure of the equivalent tree as given by the structural transfer module (in this case the structure is the same as the structure of the graph)
- the decoration of the tree nodes.

The decoration of each node lists

- the Universal Word
- the semantic relation relative to its moither node (noted as a monovalued variable RSUNL)
- the attributes of the node (noted as a multivalued variable VARUNL)
- the id number (noted as the monovalued variable INST).

3.2. Graphs containing nodes with more than one mother node

In a tree, the root node has no mother node, and the other nodes have only one mother node. This is of course generally not the case for a graph, where all the nodes (including the entry one) may have several mother nodes.

Let’s for instance consider the graph of fig. 4, where the entry node (« institute ») has a mother node (« establish ») the arc joining the first node to the second bearing the relation *obj*:

```
obj(establish(icl>found).@past,institute(ic
l>facilities).@present.@entry)
```

In order to get a tree, with a root node without mother node, the relation is inverted in the transfer module, and becomes

```
xxobj(institute(icl>facilities).@present.@e
ntry, establish(icl>found).@past)
```

where *xxobj* represents the inverse relation of the *obj* relation . The *obj* relation in the original graph expresses the fact that « institute » is the *obj* of establish, whereas the *xxobj* relation in the modofied graph expresses the fact that « establish » has « institute » as *obj.* Such an "inverted relation" is usally deconverted into French as a relative clause. The deconverted French text reads *"L'université des Nations Unie est un institut que l'Assemblée Générale des Nations Unies a fondé en 1975"."*

3.3. Graph containing a closed circuit

An equivalent tree structure of a graph containing a closed circuit may be obtained by opening the circuit, splitting one of its nodes as shown on fig.5 (the node "lecturer".splitted)

The new created node bears the same id number as the original one, indicating that it refers to the same object. In this example, this new node will be translated in French by the possessive "son", and the deconverter output reads *Le conférencier a lu son papier "*

3.4. Hypergraphs

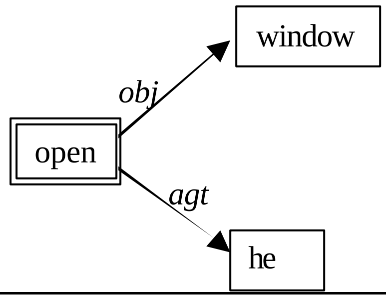
The processing of an hypergraph (graph containing subgraphs) is quite straightforward: the resulting tree is a tree containing subtrees.

English text: **He doesn’t open the window.**

Graph :

```
agt(open(icl>do).@entry.@not,he)
obj(open(icl>do).@entry.@not,window.@def)
```

Graph structure:



Output tree:

```
                                          |-- 2:'WINDOW
                                          |
                              1:'OPEN' ---!-- 3:'HE'
```

Tree decoration:

```
1 'OPEN': VARUNL(ENTRY,NOT),INST(1)
    2 'WINDOW': VARUNL(DEF), RSUNL(OBJ),INST(1)
    3 'HE': RSUNL(AGT),INST(1)
```

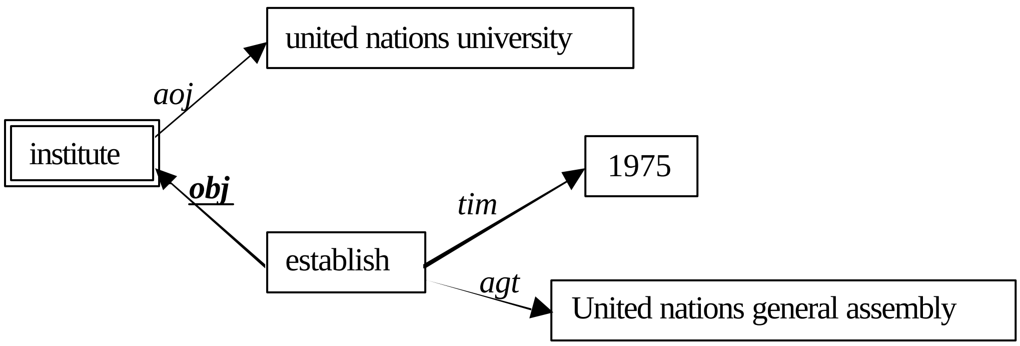
Fig 3. Structural transfer for a graph with tree structure..

English text: **The United Nations University is an institute which was founded by the United Nations General Assembly in 1975.**

Graph:

```
aoj(institute(icl>facilities).@present.@entry,united nations
university(icl>facilities))
obj(establish(icl>found).@past,institute(icl>facilities).@present.@entry)
agt(establish(icl>found).@past,united nations general
assembly(icl>organization))
tim(establish(icl>found).@past,1975)
```

Graph structure



Output tree

```
                       |-- 2:'UNITED NATION UNIVERSITY'
                       |                      |-- 4:'1975'
-- 1:'INSTITUTE' ------!-- 3:'ESTABLISH' -----!-- 5:'UNITED NATIONS GENERAL ASSEMBLY'
```

Tree decoration:

```
1 'INSTITUTE': VARUNL(PRESENT,ENTRY),INST(1)
    2 'UNITED NATIONS UNIVERSITY' RSUNL(AOJ),INST(1)
    3 'ESTABLISH': UL('<ESTABLISH>'), VARUNL(PAST), RSUNL(XXOBJ),INST(1)
      4 '1975':  RSUNL(TIM),CAT(CATCARD)
      5 UNITED NATIONS GENERAL ASSEMBLY': RSUNL(AGT),INST(1)
```

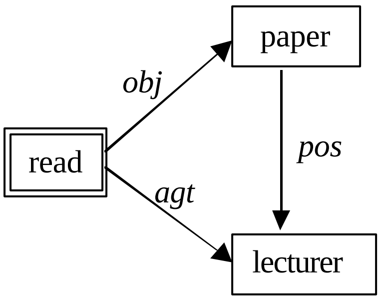
Figure 4 :Structural transfer of a graph whose entry node has a mother node

English text : **The lecturer read his paper.**

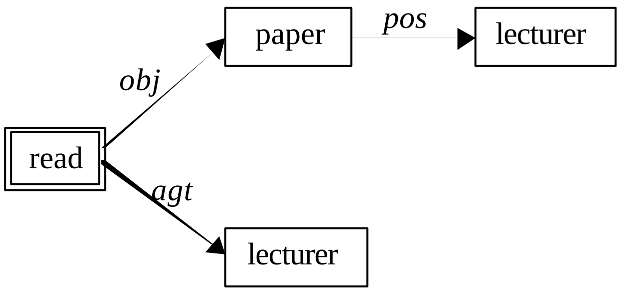
Graph :

```
agt(read(icl>do).@entry.@past,lecturer.@def)
obj(read(icl>do).@entry.@past,paper(icl>article))
pos(paper(icl>article),lecturer)
```

Graph structure:



Node splitting:



Output tree:

```
                        |-- 2:'PAPER' ----- 3:'LECTURER
                        |
          1:'READ' -----!-- 4:'LECTURER
```

Tree decoration

```
1 'READ': VARUNL(ENTRY,PAST),INST(1)
    2 'PAPER': RSUNL(OBJ),INST(1)
      3 'LECTURER': RSUNL(POS),INST(1)
    4 'LECTURER': VARUNL(DEF), RSUNL(AGT),INST(1)
```

Figure 5. Structural transfer for a graph containing a closed circuit.

4. UNL graph to NL tree lexical transfer

The structure of the UNL universal words makes in principle the lexical transfer a straightforward process.

A Universal Word like *mouse(icl>animal)* comprises indeed an headword "*mouse*" and a restriction "*icl>animal*" whose aim is to disambiguate the UW : distinction between *mouse(icl>animal)* and *mouse(icl>device)*.

But in practice incompletness or inadequacies of the dictionaries leads either to use a treatment of the unknown word or an interactive lexical transfer.

4.1. Treatment of the unknown word

The treatment of the unknown words (that is of Uws whose NL language equivalents are not available in the dictionaries) may be based on the restriction of the UW and/or on the semantic relations the UW participates to.

4.1.1. Treatment of the unknown word based on the UW restriction

Using the restriction of the UW, we perform a partial treatment of the unknown word: the UW is not translated (the headword appears in the deconverted sentence), but the sentence is as far as possible correctly build.

This is shown on figure 6 where the graph contains two UWs supposed unknown. Testing the restrictions of the unknown UWs `rake(icl>do)` and `rake(icl>thing)` indicates that the first one is a verbal concept, the second one a thing concept, which allowed a correct construction of the sentence.

English text : **He rakes the leaves with the big rake.**

Graph :

```
agt(rake(icl>do).@entry,he)
obj(rake(icl>do).@entry,leaf(fld>bo
tany).@def.@pl)
ins(rake(icl>do).@entry,rake(icl>th
ing))
mod(rake(icl>thing),big(mod<thing))
```

French output text : **Il <<rake>> les feuilles avec le? grand? <<rake>>.**

Fig 6 Treatment of the unknown word based on the UW restrictions

4.1.2. Treatment of the unknown word based on the semantic relations

The semantic relations may also be used to determine the nature of the unknown word, allowing thus to obtain the correct sentence structure.

Figure 6 shows the deconversion result for a (unrealistic) graph where two unknown UWs without restrictions are present : *rake:01* and *rake:02* (the two different ids :01 and :02 indicate that these UWs are associated to two different nodes).

The different natures of both UWs were determined by using the semantic relations: the first instance of the UW rake, being the origin of an *agt* relation, was considered as a verbal concept, while the second one, being the target of an *ins* relation, was considered as a nominal concept.

English text **He rakes the leaves with the big rake.**

Graph : `agt(rake:01.@entry,he)`
`obj(rake:01.@entry,leaf(fld>botany).@def.@pl)`
`ins(rake:01.@entry,rake:02)`
`mod(rake:02,big(mod<thing))`

French output text: **Il <<rake>> les feuilles avec le? grand? <<rake>>.**

Fig 6 Treatment of the unknown word based on the semantic relations.

4.2. Interactive lexical transfer

Our local deconverter may work in an interactive lexical mode. In this mode, for each UW in the graph, the French equivalent(s) present in the dictionaries are displayed for choice (figure 7).

```
Meeting(icl>event)
Click on one item below
-----------------------
Entering a new equivalent

meeting(icl>event)
réunion
CAT(CATN),GNR(FEM)

meeting(icl>event)
rencontre
CAT(CATN),GNR(FEM)
```

Figure 7 : Interactive lexical transfer

If no satisfactory equivalent is present in the dictionaries, the user may enter the correct equivalent, which is stored in an auxiliary dictionary, and becomes immediately available.

This interactive mode makes use of the PARAX-UNL hypertextual multilingual database (Blanc 1999)

5. Argument transfer

By argument transfer, we mean the relation between a UNL semantic relation and the corresponding syntactic function in the target natural language. It is not a one to one relation.

We will show here on an example how testing the restriction of a predicate may help finding the syntactic function associated to a semantic relation.

In the UNL language, one distinguishes the verbal concepts *do*, *occur*, *be*. For instance, the graph of fig. 8 contains the UW « open(icl>do) », whereas the graph of fig. 9 below contains the UW « open(icl>occur) ».

Both UWs are translated into French by the same verb, « ouvrir » (or in English by the same verb « to open »). But it is clear that in the case of «open(icl>do) », the subject syntactic relation for the French (or the English) verb corresponds to the *agt* relation (figure 8), but to the *obj* relation in the case of the « open(icl>occur) » UW.

That means that in such a case the restriction had to be tested in order to find the subject of the sentence.

He doesn't open the window.
`agt(open(icl>do).@entry.@not,he)` `obj(open(icl>do).@entry.@not,window.@def)`
Il n'ouvre pas la fenêtre.

Figure 8 The obj relation of this graph corresponds to the syntactic object relation in French or English

The window doesn't open.

`[S]`
`;<SUZHOU_4>`
`obj(open(icl>occur).@entry.@not,window.@def)`
`[/S]`

La fenêtre n'ouvre pas.

Figure 9 The obj relation of this graph corresponds to the syntactic subjet relation in French or English

6. Conclusion

Such a UNL graph to Natural Language tree transfer proved to be quite feasible, and allowed us to reuse an existing French generator.

7. References

Boitet C. 1997 GETA's methodology and its current developments *PACLING'97, Meisei University, Ohme, Japan, sept 97, Proceedings 23-57.*

Blanc E (1999) PARAX-UNL, a large scale multilingual hypertextual database. *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium 1999 (NLPRS 99), pp 507-510. Tsinghua University Press, Beijing 1999.*