A UNL Deconverter for Chinese

Xiaodong Shi, Yidong Chen

Institute of Artificial Intelligence
School of Information Sciences and Technologies, Xiamen University
361005 Xiamen, China
{mandel, ydchen}@xmu.edu.cn

Abstract. This paper describes the internal working of a novel UNL converter for the Chinese language. Three steps are involved in generating Chinese from UNL: first, the UNL expression is converted to a graph; second, the graph is converted to a number of trees. Third, a top-down tree walking is performed to translate each subtree and the results are composed to form a complete sentence. Because each node is visited exactly once, the algorithm is of linear time complexity and thus much faster than the standard deconverter provided by the UNL center. A manual evaluation effort was carried out which confirmed that the quality of the Deconverter output was better than that of the standard deconverter.

1 Introduction

Although the UNL [1],[2] center provides a language independent generator [3] which can deconvert UNL expressions into any language provided that a UW dictionary, a set of deconversion rules, and optionally a co-occurrence dictionary are available for that language, that deconverter has a number of deficiencies: First, the deconversion rules are rather difficult to write because of the cryptic formats imposed by the deconversion specification. Second, although the power of the deconverter is claimed to be that of the Turing machine [4], its speed is rather slow and thus unsuitable for the main web application, embedded multilingual viewing of a UNL document that is one of the key goals of the UNL. Third, most importantly, the deconversion software is not open-sourced, so that fixing any bugs or introducing much-needed improvements is at the mercy of the UNL center, which has been rather lacking in technical support and in releasing new versions. So we think it is necessary to develop our own deconverter for Chinese. This paper describes such an endeavor. However, it should be noted that although we concentrate on generating Chinese from the UNL expressions, nothing in our deconverter is inherently related to Chinese, thus the deconverter is also language independent.

This paper is organized as follows: Section 2 will describe the main components of the deconverter and the algorithms involved. Section 3 will focus some issues in generation, especially those related to the Chinese language, and in Section 4 we will briefly discuss related work in the literature. In Section 5 we will give example uses of the deconverter and finally we will present the conclusions.

Universal Network Language: Advances in Theory and Applications.
Research on Computing Science 12, 2005, pp. 167–174.

2 The Main Components of the Deconverter

The deconverter has three components: the first converts a UNL expression into a graph, the second breaks the graph into a number of trees, and in the third, a recursive top-down tree walking is performed to translate each subtree and the results are composed to form a complete sentence. They will be described in more detail below.

2.1 Graph Construction

Converting a UNL expression into a graph is straightforward. In this respect, the list form of a UNL expression is simpler to convert than the normal form:

```
{unl}
[W]
language(icl>abstract thing).@present.@entry:00
UNL(icl>language).@topic:01
common(aoj>thing):02
use(icl>do).@present:03
communication(icl>action).@pl:05
network(icl>thing):06
[/W]
[R]
00aoj01
00mod02
03obj00
03pur05
05mod06
[/R]
{/unl}
```

Fig. 1. The list form of a UNL expression.

In essence, the normal form is converted into the list form, from which a *directed* graph is constructed. The nodes correspond to UWs in the UNL expression and the edges of the graph are labeled with relations, pointing from head to the dependents (if a relation is of the form UWID1 **rel** UWID2, then UWID1 is the head, and UWID2 is the dependent). In the case of a compound UWID, a node corresponding to it is also created because it can have attributes attached, e.g. some of the attributes from the ITU corpus are @def, @topic, @double_slash, etc, which usually apply to all the UWs in the scope denoted by the compound UWID. It should be noted that two nodes with different IDs but otherwise identical information cannot be collapsed into one, as co-referential nodes share the same ID (or use a UW with no ID at all). One node of the graph is of central importance: the entry node.

2.2 Graph-to-tree Conversion

The second component is called the graph-to-tree conversion. If a node in the graph has **two** or more edges (or **one** or more edges for the entry node) pointing to it, then all the directed edges except one must be severed. Exactly which one is retained depends on a breath-first traversal of the graph starting from the entry node. See the following paragraph for the detailed discussion of this operation. The edges encountered in this traversal are called the forward edges and other edges will be severed, but the traversal will continue at the head of the severed edges. For the UNL expression shown in Fig. 1, the graph conversion will produce the following two trees:

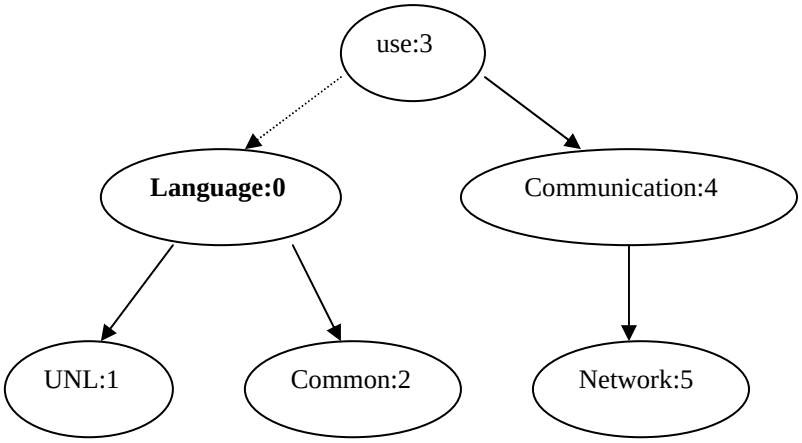

Fig. 2. The two trees converted from the graph. The severed edges are in dotted line (from node 3 to 0) and the node number indicates the graph traversal order. The edge is severed because node 0 is the entry node with an in-degree of 1.

There are two cases in the handling of a node with a severed link depending on whether it is **duplicated** or **attributive**. (The latter means that it is modified by an attributive clause, or in other words, it is an entry node of a scope, but with a governing head). And the reason for choosing this is somewhat **syntactic**. In our opinion, the introduction of the scope node also has a very strong syntactic flavor, besides making UNL more hierarchical. In general, an underspecified UNL graph can mean different things to different people when syntactic information in the original sentence is not represented with a scope node (or a Compound UW)[1]. The following UNL graph illustrates this point (slightly adapted from [12]):

[1] This observation benefited from a discussion with Dr Etienne Blanc while I was visiting GETA-CLIPS in 2004.

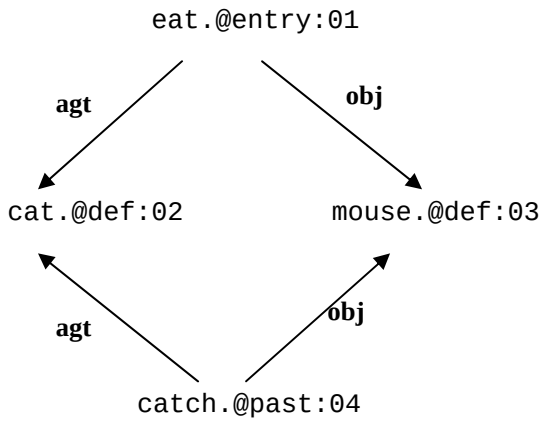

Fig. 3. A UNL graph for either "The cat eats the mouse it caught" or "The cat which caught the mouse eats it".

When node 02 is duplicated and node 03 is made attributive, we could get the deconverted result: "The cat eats the mouse it caught" (or "the cat eats the mouse the cat caught" if no pronominalization is implemented); when node 02 is made attributive and node 03 is duplicated, we could get the deconverted result: "The cat eats the mouse it caught". (The third possibility: "The cat eats the mouse. The cat caught the mouse." is not implemented, as might be the result when both nodes 02 and 03 are duplicated)

In the implementation, the severed edges are still retained in the graph, and this *conversion* is done on the fly just before generation, for optimal performance.

2.3 Natural Language Generation from the Trees

A simple hypothesis of the generation algorithm is the **compositionality**. It states that the generation of the whole tree can be constructed from its subtrees. Suppose a node V has the set of adjacent nodes: A(V), and let T be a function from the subtree dominated by V to its translation, and C be a composition function from the *subpart*s (translation of the subtree, henceforth), then we have

$$T(V)=C(W(V), T(V_1), T(V_2), T(V_3),...), \quad \cup V_i = A(V) \qquad (1)$$

where W(V) is the proper word translation of the node V.

There are several things to note during the composition process. First an ordering function **O** determines the relative orders of the subparts according to the roles played by them. Relation Labels (RL) on the tree edges provide most of the information. Then the particular choice of the head word translation W(V) may have its own subcategorization requirement and ordering constraints. So the ordering function can be specified as follows:

$$\mathbf{O}: \ W(V), RL_1, RL_2,\ldots RL_n, \rightarrow I^{n+1} \qquad (2)$$

where I is the set of integers. After the orders are determined, the subparts are simply concatenated.

Second some words (*glue* words) may have to be prepended or appended to the subparts to make case roles explicit. This is the case for Chinese. Other languages, e.g. Tamil [5], may require morphological generation. The generation algorithm can easily handle these minor divergences.

Note that translations produced by the severed nodes are also composed. These are mostly realized as attributive clauses. The node is aware of whether it is generating standalone, or as a severed subtree.

A simple way of avoiding loops in generation (because the tree conversion is a logical process) and gathering information from the already generated nodes is to mark each node as they are visited.

The deconverter is very impressive in its speed. Because each node is visited exactly once, the algorithm is of linear time complexity and thus much faster than the standard deconverter provided by the UNL center. As to the quality of the deconverter, a manual evaluation effort of the translations of a few hundred sentences was carried out, which confirmed that the quality of the output was also superior to that of the standard deconverter.

3 Some Issues in Generation

We should note that in general, the compositionality hypothesis is not always correct. For examples, in some idiomatic constructions, one subpart has to be imbedded in another subpart. And although UNL encodes the natural language sentences in a language neutral way, its particular choice of UWs is inevitably influenced by the English language, which is used as a specification language to make UNL accessible to a larger audience. So some UWs may not have appropriate lexicalization in another language and so awkward translations may result.

One particular pitfall in UNL is the specification of the conjunction relation: the UNL expression is ambiguous as to whether a conjunction is at sentence level or predicate (mostly verbal or adjectival concept) level. The correct function must be inferred from other relation label which may be present. Other relation labels may also present problems, as there have been lots of argument among the language centers as to whether a particular relation is need, but as long as UNL expressions are consistent in this respect (admittedly a hard goal to attain), no serious problem should follow.

One important aspect of the Chinese generation is the insertion of appropriate classifier, or measure word, for nouns that can be counted. This has been notoriously difficult to handle in the original rule set developed for the standard deconverter of the UNL center, and more than one hundred rules are given, each for a different classifier. In our implementation, the classifier is treated as a *glue* word, and is generated directly from the head noun using a classifier table which can be modified separately.

Some UWs expressing prepositions in English are problematic in generating correct Chinese. They are typically realized into two Chinese words surrounding the governed noun phrases. To circumvent this problem, we introduced the concept of a

parametric attribute, which share a common prefix. One word of the Chinese translation is chosen as the main word which corresponds to the UW, the other word is expressed using a *parametric* attribute (concatenating the prefix with the second word). In generation, when a *parametric* attribute is found to be present, the second word is extracted and properly appended.

To resolve the problem of ambiguity in lexicalization (as is the case for near-synonyms), a co-occurrence dictionary is also used, but these data are collected automatically from a huge Chinese corpus with more than 8 billion Chinese characters. Since automatic parsing for Chinese is still not very reliable, we only extract bigrams using **segtag**, a segmentation and tagging program developed by the Center for Language Technology[2] of Xiamen University, which can be downloaded for free from http://clt.xmu.edu.cn.

4 Related Work on UNL Deconverter

Although there are 15 language centers listed on the UNL Universe [6], only a few deconverters are working, e.g. the Russian, French, and Spanish ones. And the details on their implementation are scanty and scarce.

[7] describes a UNL to French deconverter which also utilizes a graph to tree converter. The tree output is then fed into an Ariane-G5 transfer program to reuse the MT facility. The generation is much more complex than described here.

[5] describes the UNL to Tamil deconverter which focus more on the syntactic and morphological generation**.**

[8] describes a deconverter for Chinese. That work was done by the Chinese center before the authors came to work there. It is because of the deficiency of that deconverter that the need for a new one is called for.

The main characteristics of the deconverter described here is its simplicity, speed and effectiveness. We think that it can be applied to other natural languages to the equal benefit exhibited by Chinese and so we make it available for download from our website http://ai.xmu.edu.cn/. Interested parties can contact the authors for the source code.

5 Deconverter in Use

The deconverter is developed on Microsoft Windows platform. We have built an IDE integrating enconversion, deconversion and UW editing. [9] describes the technical aspects of both the deconverter and enconverter in more detail. The following figure shows the IDE interface.

[2] The Institute of Artificial Intelligence is a major research branch of the Center for Language Technology, which also includes faculty from the Humanities departments.

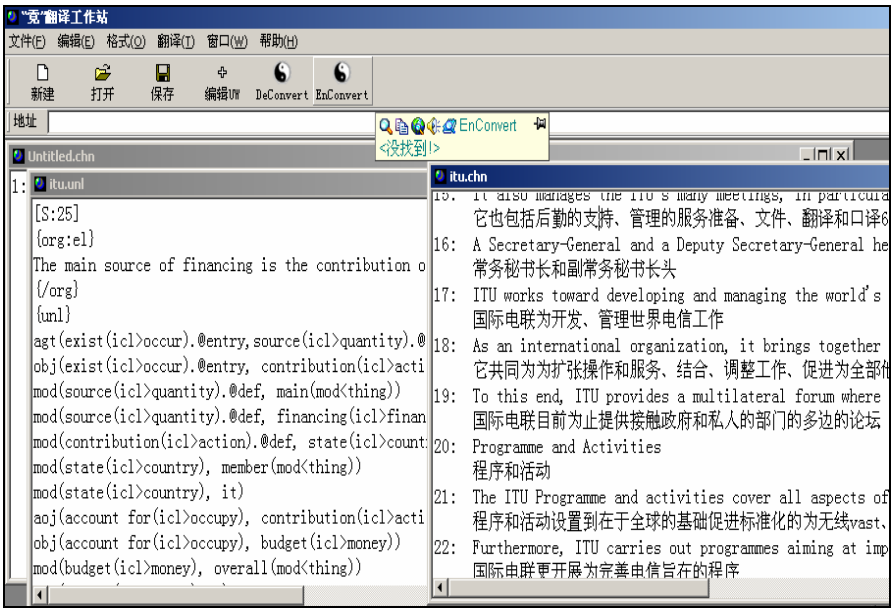

Fig. 3. The UNL enconversion and deconversion IDE.

We have also built a language server [10] (not in the sense of the UNL center), which is implemented as a SOAP-compliant Web Service [11]. The server runs as a Apache module and effectively offers a cross-platform Remote Procedure Call (RPC) for enconversion and deconversion. It also provides an RPC endpoint for MT via UNL. If other enconverters or deconverters were available, MT would be available for those languages.

6 Conclusions

This paper describes our implementation of a UNL deconverter for Chinese with graph construction, graph-to-tree conversion, and recursive top-down generation as three components. It's very fast and gives better performance than the standard deconverter with a Chinese rule set. Although it is developed with Chinese in mind, it is also a language neutral software and thus can be used for other languages.

Acknowledgements The work described in this paper was mostly done while the authors were in the Chinese Language Center, the Graduate School of the Academy of Chinese Sciences, in the year 2001. The authors wish to express thanks to Professor Wen Gao, the director of the Chinese Language Center, and Dr Hiroshi Uchida, of the UNL Center in Tokyo, for the support provided.

We would also like to thank the anonymous reviewer for very constructive comments on the first version of this paper.

This work is partially sponsored by China high-tech program grant No. 2001AA114103 and grant No. 2002AA117010.

References

1. UNU/IAS/UNL Center: The Universal Networking Language (UNL) Specifications 3.0, August 2000
2. Martins R.T., Rino L.H.M., Nunes M.G.V., Montilha G., Oliveira Jr. O.N.: An interlingua aiming at communication on the Web: How language-independent can it be? Workshop on Applied Interlinguas: Practical Applications of Interlingual Approaches to NLP. April 30, 2000. Seatle, Washington, USA.
3. UNL Center, Deconverter Specifications, Version 2.5, UNL-TR-2001-001
4. http://www.undl.org/unlsys/public/deco.html
5. Dhanabalan T., Geetha T.V.: UNL Deconverter for Tamil. International Conference on the Convergences of Knowledge, Culture, Language and Information Technologies, December 2003, Alexandria, EGYPT
6. UNL Resources by ISI, http://cui.unige.ch/isi/unl/
7. Blanc É. & Sérasset G.: From Graph to Tree: Processing UNL Graphs using an Existing MT System. Proc. The first UNL open Conference, Suzhou, China, 22-26 November 2001
8. Xiong, W.X: Some Problems on Machine Translation via Interlingua, Applied Linguistics. 3 (1998) 69-75
9. Shi X.D.: UNL enconversion and Deconversion Based on the Neon MT system. Technical Report, Chinese Language Center, 2002.
10. Shi X.D.: The UNL Language Server for Chinese, Chinese Language Center, 2002.
11. Curbera F., Nagy W., and Weerawarana. S.: Web services: Why and how. In Workshop on Object Orientation and Web Services OOWS2001, 2001
12. Sérasset G., Blanc E.: Remaining Issues that could Prevent UNL to be Accepted as a Standard, Convergences 2003, Alexandria, Egypt, 26 December 2003.