# Generative and Enumerative Lexicons in the UNL Framework

Ronaldo Martins
UNDL Foundation
Geneva, Switzerland
r.martins@undlfoundation.org

Vahan Avetisyan
UNDL Foundation
Geneva, Switzerland
v.avetisyan@undlfoundation.org

## ABSTRACT
This paper presents an ongoing work related to the development of natural language processing systems to be used in the UNL framework. We address the problem of dealing with irregular forms in the current UNL-driven dictionaries, and claim that they are mainly derived from the structure of existing UNL engines (EnCo and DeCo). We propose a new dictionary architecture which incorporates several enhancements in order to assure flexibility and scalability to the development of UNL language resources.

## Keywords
UNL, dictionary, lexicon, natural language processing.

## 1. INTRODUCTION
The Universal Networking Language (UNL) [1,2] is a knowledge representation language that can be used for several different tasks in natural language engineering, such as machine translation, multilingual document generation, summarization, information retrieval and semantic reasoning. It has been originally proposed by the Institute of Advanced Studies of the United Nations University, in Tokyo, and has been currently promoted by the UNDL Foundation, in Geneva, Switzerland, under a mandate of the United Nations.

In the UNL approach, the information conveyed by natural language is represented, sentence by sentence, as a graph whose nodes represent concepts and whose edges represent binary semantic relations between concepts. The nodes are called Universal Words (or simply UWs) and can be modified by a predefined set of attributes which cover the information that cannot be represented as UWs or relations. The set of relations is also predefined in the UNL Specifications and consists of 46 semantic cases (such as agent, object, instrument, etc) that are claimed to be language independent, as well as the set of UWs and their attributes.

In the UNL framework, the process of representing information into UNL is called "enconversion", and the process of extracting natural language sentences out of UNL is called "deconversion". These special names reflect the idea that those processes may not be exactly coincident with natural language analysis and generation, and that may involve specific strategies. Additionally, both processes are supposed to be performed by language independent engines (EnCo and DeCo, respectively), to be parameterized with natural language grammars, dictionaries and other lexical repositories (such as the knowledge base and the co-occurrence dictionary).

Even though the "enconversion" and "deconversion" processes are peripheral to the essence of UNL, which is to represent knowledge in a machine-tractable way, the fact is that the current development of UNL has been governed by the algorithms for UNLization and deUNLization, which actually state the limits of what can be enconverted into and deconverted from UNL. In that sense, the architecture of the UNL system has been of critical importance, and defines the present agenda of the UNL initiative.

In what follows, we investigate one of the main problems in the current architecture of EnCo and DeCo: the structure of the dictionaries, particularly with reference to the treatment of irregular forms. In the next section, we present the problem in the context of English and two highly-inflected languages (French and Portuguese) and analyze the available solutions in the current UNL dictionary structure. Section 3 sketches an alternative, which has been implemented in two other engines (EUGENE and IAN), still under development. And the final section concludes with some remarks on the implications of such an architectural change.

## 2. THE PROBLEM
One of the most long-standing assumptions in language description is that languages are made up of words, and that words can be inter-related in several different ways as to form more comprehensive sets, such as lexical networks, and inflectional and declension paradigms. Indeed, the possibility of grouping words according to their internal (morphological) structure has appeared, in the Western tradition, in the 4th century [3], and ever since has been adopted as a main strategy for both teaching languages and describing their morphology, as indicated in Table 1.

Table 1. Sample of regular verbs from English, French and Portuguese

| | |
|---|---|
| English | I admire, you admire, he admires, we admire, you admire, they admire<br>I analyze, you analyze, he analyzes, we analyze, you analyze, they analyze<br>I approve, you approve, he approves, we approve, you approve, they approve |
| French | J'admire, tu admires, il admire, nos admirons, vous admirez, ils admirent<br>J'analise, tu analises, il analise, nos analisons, vous analisez, ils analisent<br>J'approuve, tu approuves, il approuve, nos approuvons, vous approvues, ils approuvent |
| Portuguese | Eu admiro, tu admiras, ele admira, nós admiramos, vós admirais, eles admiram<br>Eu analiso, tu analisas, ele analisa, nós analisamos, vós analisais, eles analisam<br>Eu aprovo, tu aprovas, ele aprova, nós aprovamos, vós aprovais, eles aprovam |

From Table 1, which covers a very small set of verbs in English, French and Portuguese in the present tense, it is relatively easy 1) to depict the stem of the word and 2) to provide rules for generating the inflected forms. In that sense, the production of resources for natural language analysis and generation is rather inexpensive and accurate, as indicated in Table 2.

Table 2. Conjugation table for the verbs appearing in Table 1

| PERSON | PRESENT | | |
|---|---|---|---|
| | ENGLISH | FRENCH | PORTUGUESE |
| I | STEM + Ø | STEM + "e" | STEM + "o" |
| You | STEM + Ø | STEM + "es" | STEM + "as" |
| He | STEM + "s" | STEM + "e" | STEM + "a" |
| We | STEM + Ø | STEM + "ons" | STEM + "amos" |
| You | STEM + Ø | STEM + "ez" | STEM + "ais" |
| They | STEM + Ø | STEM + "ent" | STEM + "am" |

Accordingly, the production of enconversion rules for EnCo and of deconversion rules for DeCo can be equally direct, as indicated below for the generation of Portuguese.

```
: {VER,P01,1PS,PRESIND,!inflect:-!inflect::}   "[o]:::"        P5;
: {VER,P01,2PS,PRESIND,!inflect:-!inflect::}   "[as]:::"       P5;
: {VER,P01,3PS,PRESIND,!inflect:-!inflect::}   "[a]:::"        P5;
: {VER,P01,1PP,PRESIND,!inflect:-!inflect::}   "[amos]:::"     P5;
: {VER,P01,1PP,PRESIND,!inflect:-!inflect::}   "[ais]:::"      P5;
: {VER,P01,1PP,PRESIND,!inflect:-!inflect::}   "[am]:::"       P5;
```

The thorough understanding of the rules above involves some acquaintance with DeCo's syntax, which can be briefly described as follows: there are two generation windows, the left and the right, and two possible actions, insertion and modification, the latter represented by {} and the former by "". Additionally, each window has four different fields CONDITION:ACTION:RELATION:ROLE, separated by colons. This means that the reading of the first rule should be the following: insert the string "o" to the right of an existing string that brings the features "VER" (verb), "P01" (paradigm 01), "1PS" (first person), "PRESIND" (present of indicative) and "!inflect"(should be inflected), and remove, from this string, the feature "!inflect" (to avoid infinite loops). The "P5" at the end of the rule indicates its priority concerning the other existing rules.

As for the UNL-NL dictionary, the six different possible forms of the verb should be stored as a single entry, which brings evident advantages concerning dictionary volume and maintenance:

```
[admir]    {}    "admire"    (VER,P01)    <p,0,0>;
[analis]   {}    "analyze"   (VER,P01)    <p,0,0>;
[aprov]    {}    "approve"   (VER,P01)    <p,0,0>;
```

Unfortunately, however, languages are not made out of regular words only. There are several different types of anomalies concerning the morphology of words that pose more than a few difficulties in the production of language resources. Some are easy to cope with, as the presence of local allographs (graphical variants for the same phoneme), but others, as radical changes in the stem or the absence of certain inflective possibilities, which is the case of defective verbs, may lead to either single-instance paradigms or may require severe backtracking. This is, for instance, the case of the verb "to be" (and quite a lot of others) in the three languages referred to above, as indicated in Table 3.

Table 3. Sample of irregular verbs from English, French and Portuguese

| | |
|---|---|
| English | I am, I was<br>I go, I went<br>I do, I did |
| French | Je suis, je fus (j'étais)<br>Je vais, j'allais (je suis allé)<br>Je fais, je fis (j'ais fait) |
| Portuguese | Eu sou, eu fui<br>Eu vou, eu fui<br>Eu faço, eu fiz |

It is not possible to derive a straight set of rules to generate the forms of the verbs "to be", "to go" and "to do" in English, French and Portuguese, especially considering tenses other than the present of indicative. In most cases, even the stem is completely different.

Given the current architecture of EnCo and DeCo we have two different possibilities to address such irregularities:

a) To consider the stem to be empty and to create single-instance paradigms; or
b) Not to analyze the verb and to store all variants in the dictionary.

In the first case, the dictionary structure will be preserved and the burden will be put on the grammar, which will bring several different paradigms, with several different rules each. This has been the option for the Portuguese dictionary structure, which has 78 different paradigms (or 5,392 rules) only to deal with the morphology of verbs[4].

In the second case, the grammar will be spared, but the dictionary will be extended in order to comprise several different variants for the same lemma. This has been the option for the English dictionary structure, whose verbal morphology is far much less complex, and which brings "be", "am", "is", "are", "was" and "were" as different entries[5]. On the other hand, the English grammar has several different backtracking rules. Given the fact that morphology rules are applied at late stages in sentence generation, this strategy can degrade considerably the efficiency of the processing.

As a matter of fact, the two options summarize the existing options in dictionary engineering. The first, which relies on the principle that "the smaller the better", can be interpreted as an early attempt to make dictionaries as generative as possible, in the sense that they should bring only base forms (lemmas) and generation rules for providing the inflections. Its main advantage concerns access (the word retrieval process is supposed to be faster), storage (it requires a smaller amount of memory space) and maintenance (changes are automatically propagated to all instances of a given entry). The second option, which is closer to the enumerative approach, states that irregular forms do not need to be artificially analyzed and regularized, and that they can be more accurately retrieved as a single atomic entity instead of a combination of several different morphemes, what is particularly true for natural language analysis. Its main advantages concern word matching (faster and more precise as there is no possibility of over-generation) and construction (it is easier and often less expensive to list the irregular forms instead of trying to define paradigms for them).

It is very difficult, however, to evaluate the dictionary architectures in isolation. Several other variables should be considered, as the morphology of the language (the number and the frequency of the irregular forms), the structure of the grammar (size, costs of elaboration and maintenance) and the architecture of the system (time and memory available for compiling rules and retrieving entries in the dictionary). In any case, it seems that the strategies adopted so far inside the UNL framework have not been very suitable for either enconversion or deconversion, and that they can be enhanced by some improvements in the dictionary and grammar structure, which are presented in the next section.

# 3. A TEMPTATIVE SOLUTION

To overcome the current shortcomings, we have been proposing some amendments to the syntax of the entries of UNL dictionaries in the scope of two projects carried out inside the UNDL Foundation: EUGENE (dEep-to-sUrface language GENErator), a new deconverter based on a high-level linguist-driven three-layered formalism; and IAN (Interactive language ANalizer), a new human-aided enconverter engine, based on the same formalism as EUGENE.

The new UNL Dictionary structure preserves the same seven fields of the old one [6]:

```
[NLW]    natural language lexical item
{ID}     identification (for indexing)
"UW"     Universal Word
```

| | |
|---|---|
| (FL) | list of features |
| L | language flag |
| F | frequency (from 0 to 255, used for enconversion) |
| P | priority (from 0 to 255, used for generation) |

In that sense, existing resources are still fully supported, and no re-work or changes in syntax are actually required. The changes affect four fields: NLW, UW, FL and L. The change concerning the field "L" is rather secondary: we have simply decided to use the two-character system proposed by ISO 639-1 to represent the names of languages.

The changes concerning the field NLW are more expressive. In the former approach, the field could contain any simple string of characters, which could correspond to bound morphemes ('s', 'ed', 'chang'); to free morphemes ('table', 'computer'); to compound words ('first-aid'); to complex words ('machine translation'); or to multiword expressions ('all bark and not bite'). We have kept all these possibilities, and added two other:

a) complex entries, such as [[switch] [off]]; and
b) regular expressions, such as [colo(u?)r] and [chang(.)].

Complex entries are necessary for generating collocations and other separable words such as many English phrase verbs. It avoids the problem of representing entries like "switch off" as "switch", and "let through" as "let", and generating the particles through the grammar, which has been the current practice, given the noticeable limitations of EnCo and DeCo to deal with infixation.

Regular expressions increase the flexibility of the NLW field. When appearing in the NLW, they are used for enconversion, as they allow for variation in the string of characters by means of wildcards, and avoid the necessity of proliferating entries that should be treated as a single one.

Regular expressions may also appear in the UW field to simplify the deconversion process, especially when dealing with named entities such as dates as proper names that should figure as temporary UWs (i.e., UWs not to be included in the dictionary). This is the case of the entry below, which should be used, for instance, to deal with dates in the format "dd/mm/yyyy":

[RegEx] {} "(0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[012])/([1-9]\d{0,2}|[1-2]\d{3})" (DATE) <,0,0>;

Finally, there are changes concerning the field FEATURE LIST, which have been remarkable as well. The modifications here concern:

a) the use of a unified tagset ("NOU", for noun; "MCL", for masculine; etc);
b) the definition of feature as an attribute-value pair, such as POS=NOU, GEN=MCL, NUM=SNG;
c) the creation of generative rules, such as NUM(PLR):=> "s", or FLX(1PS&ET0&IND)=:"am".

The adoption of a unified tagset has been derived mainly from the need of standardizing the language resources inside the UNL framework. The idea is to move gradually to a more collaborative environment, where resources would be freely and intensively reused and exchanged, and this would never be possible if dictionaries continued to be rather subjective and authorial. Even though the decision may incur, at a first glance, in same sort of reductionism, because it would impose the same theoretical terminology to all participants in a multilateral and transnational project, the claim for convergence in language engineering is not new, and has been pursued, for instance, by the Lexical Markup Framework[7], proposed by the International Standards for Language Engineering (ISLE) Project, the successor of the EAGLES initiative, which discussed several strategies for ensuring reusability and interoperability between lexica and corpora produced in Europe. Additionally, the use of a single metalanguage among the participants in the project is a necessary condition for guaranteeing the language independency of the set of UNL attributes (such as "@pl", "@past", etc), which is largely dependent on descriptive morphology.

The second change is, once again, a step towards generality (and scalability) in UNL. Contrarily to EnCo and DeCo, which only admit constants, both EUGENE and IAN allow for variables, an indispensable feature for abridging natural language grammars and making them easily maintainable. Treating features as attribute-value pairs leads to very general rules such as:

(PER, >per)&(^PER):=(->per)&(+PER);

which, in EUGENE's and IAN's syntax, means that the value of the attribute 'PER' (person) should be transferred to the right node if there is a person agreement condition between them (expressed by '>per'). The same rule, in EnCo's and DeCo's syntax, would have to be written as 6 different rules, one for the each of the values of PER[8]:

```
:{PER,1PS,>per:->per::} {^PER:+PER,+1PS::} P10;
:{PER,2PS,>per:->per::} {^PER:+PER,+2PS::} P10;
:{PER,3PS,>per:->per::} {^PER:+PER,+3PS::} P10;
:{PER,1PP,>per:->per::} {^PER:+PER,+1PP::} P10;
:{PER,2PP,>per:->per::} {^PER:+PER,+2PP::} P10;
:{PER,3PP,>per:->per::} {^PER:+PER,+3PP::} P10;
```

Finally, the last change concerns the introduction of generative rules inside the dictionary to cope with the irregular forms referred to in the last section. The main purpose is to make the dictionary as enumerative as possible, but not to transfer the responsibility for generating irregular forms to the grammar. Our solution was to register the irregularities in the dictionary itself, not as separate entries, but rather as a generative schema that could be triggered by the grammar.

For the sake of an example, let us consider, once again, the case for generating the present of the indicative of the verb 'to be' in French ("être").

a) Generative (DeCo)

```
Dictionary
[] {} "be" (VER) <e,0,0>;
Grammar
:{[[be]],1PS:::} "[suis]:::" P5;
:{[[be]],2PS:::} "[es]:::" P5;
:{[[be]],3PS:::} "[est]:::" P5;
:{[[be]],1PP:::} "[sommes]:::" P5;
:{[[be]],2PP:::} "[êtes]:::" P5;
:{[[be]],3PP:::} "[sont]:::" P5;
```

In the DeCo's generative approach, the dictionary brings one single entry (which is actually empty) and the verb is entirely generated from the grammar.

b) Enumerative (DeCo)

```
Dictionary
[suis] {} "be" (VER,1PS) <e,0,0>;
[es] {} "be" (VER,2PS) <e,0,0>;
[est] {} "be" (VER,3PS) <e,0,0>;
[sommes] {} "be" (VER,1PP) <e,0,0>;
```

```
[êtes] {} “be” (VER,2PP) <e,0,0>;
[sont] {} “be” (VER,3PP) <e,0,0>;
Grammar
?{[suis],^1PS:::} {:::} P5;
?{[es],^2PS:::} {:::} P5;
?{[est],^3PS:::} {:::} P5;
?{[sommes],^1PP:::} {:::} P5;
?{[êtes],^2PP:::} {:::} P5;
?{[sont],^3PP:::} {:::} P5;
```

In the enumerative approach, the verb is entirely generated from the dictionary, but the grammar brings 6 backtracking rules (they are preceded by “?”), which can considerably delay the process if the correct option would be the 3PP (third person of plural). In this case, the system would backtrack five times before retrieving the exact match.

c) Generative (IAN and EUGENE)

```
Dictionary
[être] {} “be” (POS=VER, PER(1PS):=”suis”, PER(2PS)=”es”, PER(3PS)=”est”, PER(1PP)=”sommes”, PER(2PP)=”êtes”, PER(3PP)=”sont” <en,0,0>;
Grammar
({1PS,2PS,3PS,1PP,2PP,3PP},^PER):=(!PER,+PER);
```

The comparison is clear: the dictionary line in case of IAN and EUGENE is longer, but there is only one entry. The grammar, however, will not be overloaded, and would be, in fact, much more generic and simpler than DeCo’s, regardless of the approach.

## 4. FINAL REMARKS

If we consider that the changes indicated preserve, in general, the overall structure of the former dictionaries, and that they constitute a careful extension rather a radical revision which would require a thorough remaking, the advantages of such approach may seem now evident. It should be reinforced, however, that we are here presenting an ongoing work, which is supposed to be finished by the end of August. In that sense, we cannot provide yet final results for the comparison between DeCo, EnCo and the enconverter and deconverter engines we have been working on. For the time being, we can only attest that we have been achieving partial results that are rather promising, as they lead to a systematic simplification of the formerly difficult and user-unfriendly process of creating grammars and dictionaries in the UNL framework.

## REFERENCES


[1] Uchida, H., Zhu, M. and Della Senta, T. (1999) A gift for a millennium, IAS/UNU, Tokyo.

[2] Uchida, H., Zhu, M. and Della Senta, T. (2005). Universal Networking Language, UNDL Foundation, Geneva.

[3] Robert Henry Robins. (1997). A Short History of Linguistics, London, Longman.

[4] Nunes, M.G.V. et alii. (1997). Developing a UNL decodifier for Brazilian Portuguese. Proceedings of the II Workshop of UNL/Brazil Project. NCE/UFRJ, Rio de Janeiro.

[5] Uchida, H. (2001). English Word Dictionary, Tokyo.

[6] UNL Center. (2001). Word Dictionary Builder. Manual. Version 2.1. Tokyo.

[7] Francopoulo G., George M., Calzolari N., Monachini M., Bel N., Pet M., Soria C. (2006) Lexical Markup Framework (LMF). LREC, Genoa.

[8] PELIZZONI, J. M. ; NUNES, M. G. V. (2005) . Flexibility, Configurability and Optimality in UNL Deconversion via Multiparadigm Programming. Research on Computing Science, México, v. 12, p. 175-194.