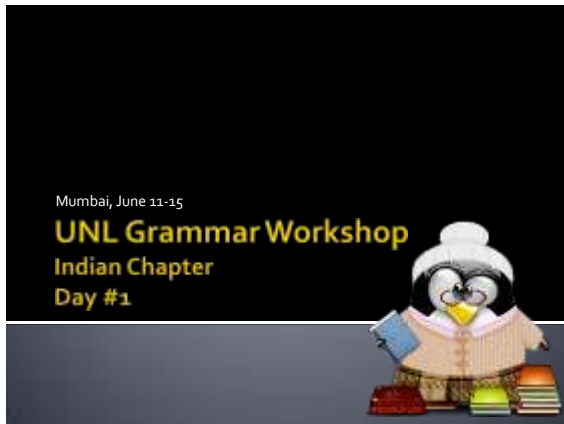




Guide

- Welcome
- Context
 - The UNL
 - The UNDL Foundation roadmap
 - The UNLiversity
 - The workshop
- Morning: First Task (Corpus)
- Afternoon: Second Task (Dictionaries)



The Universal Networking Language
(UNL)

UNL translation
knowledge representation

Universal
Networking
Language



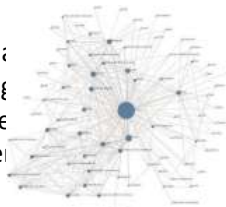
UNITED NATIONS
UNIVERSITY

1996



Foundations (I)

Informa
language
represent
independ



natural
an be
language-
structure.

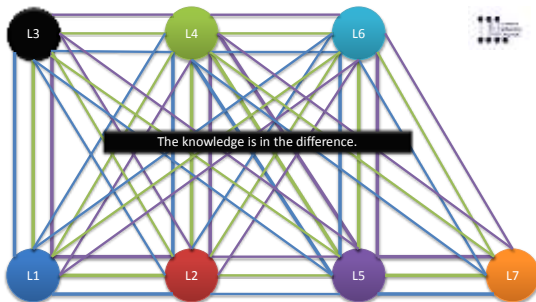
Foundations (II)

Relations between languages can be expressed in three layers:

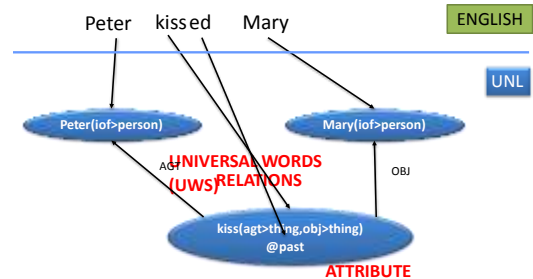
- | | |
|----------------------------|-------------------------|
| CONCEPTS | = Universal Words (UWs) |
| CONCEPT MODIFIERS | = Attributes |
| RELATIONS BETWEEN CONCEPTS | = Relations |



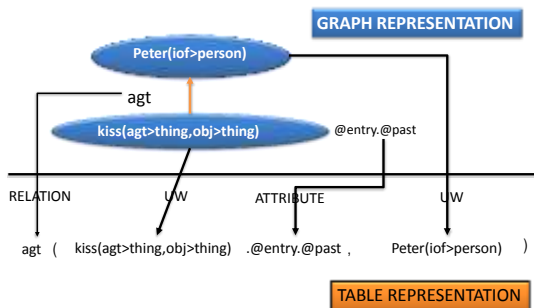
The Universal NETWORKING Language



Natural Language-to-UNL (UNL-ization)



Syntax of UNL



UNL document

```
[D]
[S:01]
(org:en)
Peter kissed Mary.
(/org)
(unl)
agt(kiss(agt>thing,obj>thing).@entry.@past, Peter(iof>person))
obj(kiss(agt>thing,obj>thing).@entry.@past, Mary(iof>person))
(/unl)
[/S]
[/D]
```

UNL (<http://anydomain/anydocument.unl>)

```
<unl>
agt(kiss(agt>thing,obj>thing).@entry.@past, Peter(iof>person))
obj(kiss(agt>thing,obj>thing).@entry.@past, Mary(iof>person))
</unl>
```



Peter kissed Mary.

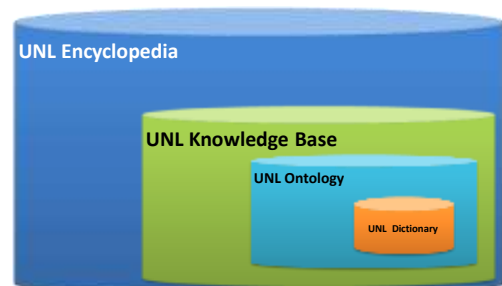


Pierre a embrassé Marie.

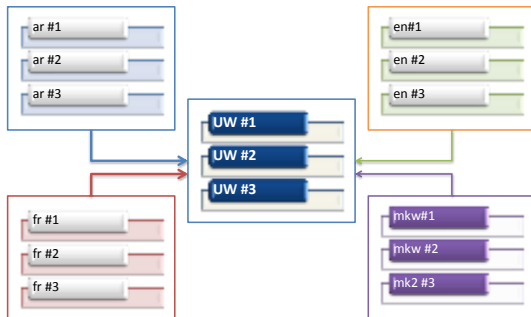


Pedro besó a María.

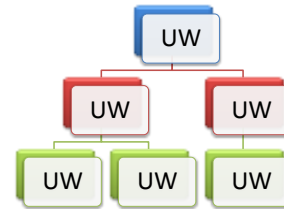
Lexical Databases



UNL Dictionary



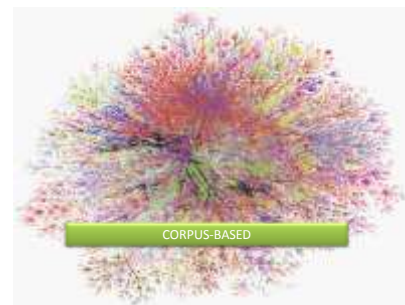
UNL Ontology



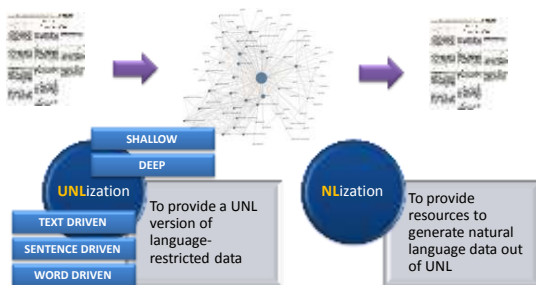
UNL Knowledge Base



UNL Encyclopedia



The UNL System



Is this translation?



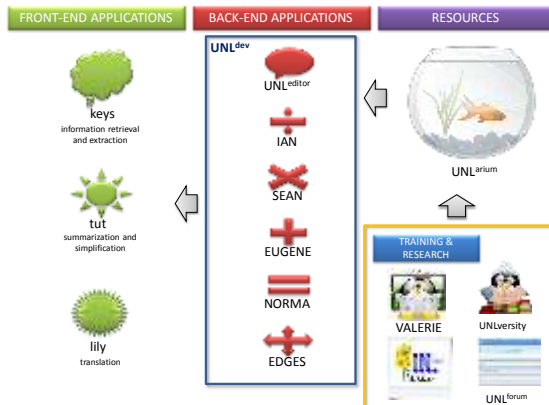
Roles of UNL

representing
organizing
retrieving
extracting
inferring
generating

monolingual
multilingual
information



the road map



UNLiversity

- ☐ UNL Symposium
- ☐ Internship Program
- ☐ UNL Grammar Workshop (A1, A2, B1, B2, C1, C2)
 - ☐ A1
 - ☐ European Chapter (Geneva, February 2012)
 - ☐ Indian Chapter (Mumbai, June 2012)
 - ☐ Asian Chapter (Macau, October 2012)
 - ☐ Middle East Chapter (Kuwait, February 2013)
 - ☐ African Chapter (Johannesburg, June 2013)
 - ☐ American Chapter (Brasilia, October 2013)
 - ☐ A2



Schedule

June 11th, 2012 - Monday
 09:00-10:00 Introduction
 10:00-12:00 I – Corpus
 14:00-17:00 II – UNL-NL dictionary
 June 12th, 2012 - Tuesday
 09:00-12:00 III – Morphology
 14:00-17:00 IV – NL dictionary
 June 13th, 2012 - Wednesday
 09:00-12:00 V – UNL-NL grammar (I)
 14:00-17:00 V – UNL-NL grammar (II)
 June 14th, 2012 - Thursday
 09:00-12:00 VI – NL-UNL grammar (I)
 14:00-17:00 VI – NL-UNL grammar (II)
 June 15th, 2012 - Friday
 09:00-12:00 Evaluation
 14:00-17:00 Discussion



Goals

- To build the basic modules of a NL-UNL (analysis) grammar
- To build the basic modules of a UNL-UNL (generation) grammar



Deliverables

During the workshop

- Experimental corpus (50 sentences)
- UNL-NL dictionary (corresponding to the corpus)
- NL dictionary (corresponding to the corpus)
- UNL-NL grammar (corresponding to the corpus)
- NL-UNL grammar (corresponding to the corpus)

Before September 15th

- Reference corpus (500 sentences)
- UNL-NL dictionary (corresponding to the corpus)
- NL dictionary (corresponding to the corpus)
- UNL-NL grammar (corresponding to the corpus)
- NL-UNL grammar (corresponding to the corpus)



Warnings

- Doubts are allowed: don't be afraid or shy.
- This is an ongoing initiative: we don't have all the answers yet.
- This is not a competition.



Participants

- Aadil Kak (Kashmiri)
- Arulmozi Selvaraj (Tamil)
- Balaji Jagan (Tamil)
- Laishram Rishikanta Meitei (Manipuri)
- Navanath Saharia (Assamese)
- Niladri Sekhar Dash (Bengali)
- Parameswarappa S (Kannada)
- Parteek Kumar (Punjabi)
- Pinkey Nainwani (Sindhi)
- Ranjan Das (Oriya)
- Renuka Devi (Telugu)
- Sachin Pawar (Marathi)
- Shailendra Kumar (Hindi)
- Trupti Nisar (Gujarati)



Corpus

Task #1: Corpus

- Translate the 50 sentences of [Corpus50_eng.txt](#) into your native language. Be as close as possible to the original.
- Save the translated text (without the English original) in a plain text (.txt) file with UTF-8 encoding and upload it to UNLWEB>UNLDEV>PROJECTS>IAN>NL FILES.
- Upload the file [Corpus50_unl.txt](#) to UNLWEB>UNLDEV>PROJECTS>EUGENE>UNL DOCUMENTS

Ronaldo Martins
UNDL Foundation

Dictionaries

Dictionaries

■ NL-UNL Dictionary (Analysis)

■ Enumerative (word forms)

```
[table] {2883} "table" (POS=NOU,NUM=SNG) <eng,0,0>;
[tables] {2883} "table" (POS=NOU,NUM=PLR) <eng,0,0>;
```

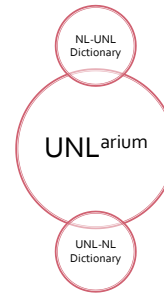
```
[foot] {2883} "foot" (POS=NOU,NUM=SNG) <eng,0,0>;
[feet] {2883} "foot" (POS=NOU,NUM=PLR) <eng,0,0>;
```

■ UNL-NL Dictionary (Generation)

■ Generative (base forms)

```
• [table] {2883} "table" (POS=NOU,NUM=SNG,PAR=M2) <eng,0,0>;
• [foot] {2883} "100284665" (POS=NOU,PAR=M2,FLX(PLR)="feet",) <eng,0,0>;
```

Building dictionaries



UNLarium



Dictionary Specs

■ Dictionary Specs

■ Dictionary structure

- a plain text file (.txt)
- one entry per line
- entries must have the following format:

```
[NLW] {ID} "UW" (ATTR , ... ) < LG , FRE , PRI > ; COMMENTS
```

[NLW]

```
[NLW] {ID} "UW" (ATTR , ... ) < LG , FRE , PRI > ; COMMENTS
```

- a multiword expression: [United States of America]
- a compound: [hot-dog]
- a simple word: [happiness]
- a simple morpheme: [happ]
- a complex structure: [[bring] [back]]
- a non-motivated linguistic entity: [g]

{ID}

```
[NLW] {ID} "UW" (ATTR , ... ) < LG , FRE , PRI > ; COMMENTS
```

- The unique identifier (primary-key) of the entry.

"UW"

[NLW] {ID} "UW" (ATTR, ...) <LG, FRE, PRI> COMMENTS

- The Universal Word of UNL. This field can be empty if a word does not need a UW.

(ATTR, ...)

[NLW] {ID} "UW" (ATTR, ...) <LG, FRE, PRI> COMMENTS

- The list of features of the NLW.
- Attributes should be separated by ",".
- It can be:
 - a list of simple features: (NOU, MCL, SNG)
 - a list of attribute-value pairs: (pos=NOU, gen=MCL, num=SNG)
 - a list of transformation rules: (plural="oo","ee")
- Replacement
 - <ATTRIBUTE>=":"<SOURCE>:"<TARGET>
 - plural:="oo":"ee"
- Left appending
 - <ATTRIBUTE>=":"<LEFT DELETION>:"<LEFT ADDITION>
 - not:="<un"
- Right appending
 - <ATTRIBUTE>=":"<RIGHT ADDITION>:"<RIGHT DELETION>
 - plural:="y>ies"

<LG, FRE, PRI>

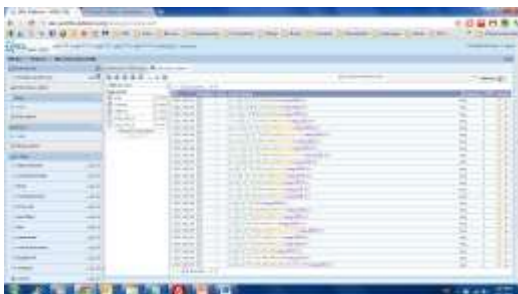
[NLW] {ID} "UW" (ATTR, ...) <LG, FRE, PRI> COMMENTS

- FLG
 - The two-character language code according to ISO 639-1.
- FRE
 - The frequency of NLW in natural texts. Used for natural language analysis (NL-UNL). It can range from 0 (less frequent) to 255 (most frequent).
- PRI
 - The priority of the NLW. Used for natural language generation (UNL-NL). It can range from 0 to 255.

Examples

```
[book]] "book" (N,NOU,SNG,M2)<eng,0,0>;
[books]] "book" (N,NOU,PLR,M2)<eng,0,0>;
[car]] "car" (N,NOU,SNG,M2)<eng,0,0>;
[cars]] "car" (N,NOU,PLR,M2)<eng,0,0>;
[dr]] "dr" (N,ABB,M0)<eng,0,0>;
[girl]] "girl" (N,NOU,SNG,M2)<eng,0,0>;
[glass]] "glass" (N,NOU,SNG,M2)<eng,0,0>;
[Bill]] "Bill" (N,PPN,SNGT,M0)<eng,0,0>;
[Edward]] "Edward" (N,PPN,SNGT,M0)<eng,0,0>;
[Geneva]] "Geneva" (N,PPN,SNGT,M0)<eng,0,0>;
[alone]] "alone" (J,ADJ,M0)<eng,0,0>;
[available]] "available" (J,ADJ,M0)<eng,0,0>;
[beautiful]] "beautiful" (J,ADJ,M0)<eng,0,0>;
[early]] "early" (A,AAV,M0)<eng,0,0>;
[never]] "never" (A,AAV,M0)<eng,0,0>;
[o'clock]] "o'clock" (A,AAV,M0)<eng,0,0>;
[arrive]] "arrive" (V,VER,TSTI,INF,M17)<eng,0,0>;
[arrives]] "arrive" (V,VER,TSTI,3PS,PRS,M17)<eng,0,0>;
[arrived]] "arrive" (V,VER,TSTI,PAS,M17)<eng,0,0>;
[arrived]] "arrive" (V,VER,TSTI,PTP,M17)<eng,0,0>;
[arriving]] "arrive" (V,VER,TSTI,GER,M17)<eng,0,0>;
```

Using dictionaries (IAN and EUGENE)



Task #2: NL-UNL Dictionary

- Extract the word list (i.e., the set of all distinct word forms) appearing in your translation of the Corpus 50
- Create the NL-UNL dictionary for all the word forms following the English model available at [English Analysis Dictionary 50](#). Use only the tags available at the [tagset](#). For further information on the dictionary structure, see [Dictionary Specs](#).
- Save the NL-UNL dictionary in a plain text (.txt) file with UTF-8 encoding and upload it to UNLWEB>UNLDEV>PROJECTS>IAN>DICTIONARIES.

Inflectional Paradigms

- Inflectional paradigms (regular) x inflectional rules (irregular)
 - book, books => inflectional paradigm (absolutely regular)
 - man, men => inflectional paradigm (irregular, but very common in several lexemes: "service man", "gentleman", "superman", etc.)
 - foot, feet => inflectional rules (irregular and uncommon)

Rule types: simple rules

- prefixation**
CONDITION := "ADDED" < DELETED;
- suffixation**
CONDITION := DELETED > "ADDED";
- infixation**
CONDITION := [REFERENCE] > "ADDED";
CONDITION := "ADDED" < [REFERENCE];
- replacement**
CONDITION := DELETED : "ADDED";
CONDITION := [INTERVAL] : "ADDED";

Rule types: complex rules

- circumfixation**
 - CONDITION := "ADDED" < DELETED , DELETED > "ADDED";
- prefixation + infixation**
 - CONDITION := "ADDED" < DELETED , DELETED > "ADDED";
- infixation + suffixation**
 - CONDITION := DELETED > "ADDED" , "DELETED" > "ADDED";

Prefixation

RULE	BEHAVIOR	BEFORE	AFTER
$X_i := "y" < "z";$	if X replace the string "z" by the string "y" in the beginning of the string	zabc	yabc
$X_i := "y" <_1;$	if X replace the first character of the string by "y"	zabc	yabc
$X_i := "y" <_0;$	if X add the string "y" to the beginning of the string	zabc	yzabc
$X_i := "y" <_i;$	if X add the string "y" to the beginning of the string (idem previous)	zabc	yzabc
$X_i := "y" < <_0;$	if X add the string "y" and a blank space to the beginning of the string	zabc	y zabc
$X_i := "y" < <_i;$	if X add the string "y" and a blank space to the beginning of the string (idem previous)	zabc	y zabc

Suffixation

RULE	BEHAVIOR	BEFORE	AFTER
$X_i := "z" > "y";$	if X replace the string "z" by the string "y" in the end of the string	abcz	abcy
$X_i := > "y";$	if X replace the last character of the string by "y"	abcz	abcy
$X_i := <_0 > "y";$	if X add the string "y" to the end of the string	abcz	abczy
$X_i := <_i > "y";$	if X add the string "y" to the end of the string (idem previous)	abcz	abczy
$X_i := <_0 > > "y";$	if X add a blank space and the string "y" to the end of the string	abcz	abcz y
$X_i := <_i > > "y";$	if X add a blank space and the string "y" to the end of the string (idem previous)	abcz	abcz y

Infixation

RULE	BEHAVIOR	BEFORE	AFTER
$X_i := [2] > "y";$	if X add "y" to the right of the second character	abc	abyc
$X_i := "y" < [3];$	if X add "y" to the left of the third character	abc	abyc
$X_i := ["b"] > "y";$	if X add "y" to the right of "b";	abc	abyc
$X_i := "y" < ["c"];$	if X add "y" to the left of "c"	abc	abyc

Replacement

RULE	BEHAVIOR	BEFORE	AFTER
X:= "y" ;	if X replace the whole entry by "y"	X	y
X:= "z" :"y";	if X replace the string "z" by "y"	azbc	aybc
X:=[2-3]:"y";	if X replace the second to the third character by "z"	abcz	ayz

Observations (I)

- Rules will only be applied if all conditions are true
 - X:="y"<"z"; ("zabc" changes to "yabc", but "abc" remains "abc" since there is no "z" to be replaced)
- String fields are necessarily continuous
 - X:="aaa"<"xyz"; ("xyzbbb" changes to "aaabbb", but "bxybyz" remains "bxybyz" since there is no continuous string "xyz" to be replaced)
- Each action is applied only once (i.e, rules are not exhaustive)
 - PLR:=o>"s"; ("X" becomes "Xs", and not "Xsssss...")
- The replacement rule applies only once to the same string
 - X:="a":**"b"**; ("aaa" becomes "baa" and not "bbb")

Observations (II)

- In prefixation and suffixation rules, the part to be deleted may be represented by the number of characters (without quotes)

PLR:="X"<**"A"**; = PLR:="X"<**1**; (ABC becomes XBC)

PLR:="XY"<**"AB"**; = PLR:="XY"<**2**; (ABC becomes XYC)

PLR:=""**">"X"**; = PLR:="o>"X"; (ABC becomes ABCX)

PLR:="C">**"X"**; = PLR:="1>"X"; (ABC becomes ABX)

PLR:="BC">**"XY"**; = PLR:="2>"XY"; (ABC becomes AXY)

Observations (III)

- In infixation rules, the position of the addition may be made with reference to the end of string by using "-".

RULE	BEHAVIOR	BEFORE	AFTER
X:=[1]>"y";	if X add "y" to the right of the first character	abc	aybc
X:=[-1]>"y";	if X add "y" to the right of the last character	abc	abyc
X:="y"<[2];	if X add "y" to the left of the second character	abcde	aybc
X:="y"<[-2];	if X add "y" to the left of the second character	abcde	abcyde

Observations (IV)

- In replacement rules, the part to be deleted may be omitted if the whole string is to be replaced
 - PLR:="ABC":**"XYZ"**; IS THE SAME AS
 - PLR:="XYZ"(ABC becomes XYZ)
- In replacement rules, the part to be deleted may be represented by an interval of characters in the format [beginning-end]
 - LR:="B":**"X"**;=PLR:=[**2-2**]:**"X"**;(ABC becomes AXC)
- The symbol **"^"** is used for negation ("**^MCL**" means "not MCL")
 - NOU&**^MCL**:="x":**"y"**; (if NOU and not MCL then replace "x" by "y")
- "<<" and ">>" add blank spaces
 - X:="a"<<"b" ("bc" becomes "a bc" and not "abc")
 - X:="a">>"b" ("bc" becomes "a bc" and not "abc")

Task #3: NL-UNL Dictionary

- Localize the UNL-NL dictionary available at [English Generation Dictionary 50](#). The localized version must reflect the word list of your translated corpus. Use only the tags available at the [tagset](#). For further information on the dictionary structure, see [Dictionary Specs](#).
- Save the UNL-NL dictionary in a plain text (.txt) file with UTF-8 encoding and upload it to UNLWEB>UNLDEV>PROJECTS>EUGENE>DICTIONARIES.

To summarize

Task #1: Corpus

- Translate the 50 sentences of [Corpus50_eng.txt](#) into your native language. Be as close as possible to the original.
- Save the translated text (without the English original) in a plain text (.txt) file with UTF-8 encoding and upload it to UNLWEB>UNLDEV>PROJECTS>IAN>NL FILES.
- Upload the file [Corpus50_unl.txt](#) to UNLWEB>UNLDEV>PROJECTS>EUGENE>UNL DOCUMENTS without doing any change to it

Task #2: NL-UNL Dictionary

- Extract the word list (i.e., the set of all distinct word forms) appearing in your translation of the Corpus 50
- Create the NL-UNL dictionary for all (and only) the word forms appearing in the corpus. In order to do that, follow the English model available at [English Analysis Dictionary 50](#). Use only the tags available at the [tagset](#). For further information on the dictionary structure, see [Dictionary Specs](#).
- Save the NL-UNL dictionary in a plain text (.txt) file with UTF-8 encoding and upload it to UNLWEB>UNLDEV>PROJECTS>IAN>DICTIONARIES.

Task #3: NL-UNL Dictionary

- Localize the UNL-NL dictionary available at [English Generation Dictionary 50](#). The localized version must reflect the word list of your translated corpus. Use only the tags available at the [tagset](#). For further information on the dictionary structure, see [Dictionary Specs](#).
- Save the UNL-NL dictionary in a plain text (.txt) file with UTF-8 encoding and upload it to UNLWEB>UNLDEV>PROJECTS>EUGENE>DICTIONARIES.