Mumbai, June 11-15

UNL Grammar Workshop

Indian Chapter

DAY #3

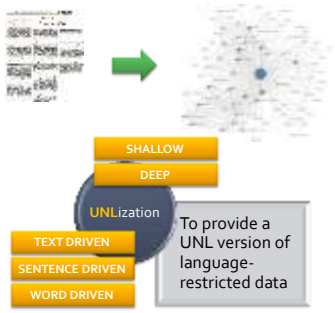
Summarizing

- Corpus
- NL-UNL (Analysis) Dictionary
- UNL-NL (Generation) Dictionary
- Inflectional grammar (for generation)

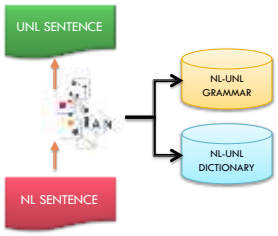
UNL-ization

natural language analysis and understanding

The UNL System



IAN



Grammar Specs

- UNL-NL Grammar Specs
  - a plain text file (.txt)
  - one entry per line
  - there are two types of rules:
    - **TRANSFORMATION RULES** are used to generate natural language sentences out of UNL graphs and vice-versa.
    - **DISAMBIGUATION RULES** are used to improve the performance of transformation rules by constraining their applicability.

Transformation Rules

Transformation Rules

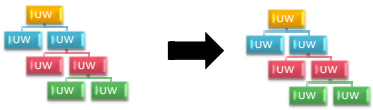
LIST-TO-LIST (LL)
Transformation Rules

ACTION	RULE
ADD	(A):=(A)(B); or (A):=(B)(A);
DELETE	(A):=-(A);
REPLACE	(A):=(B);
MERGE	(A)(B):=(C);
DIVIDE	(A):=(B)(C);

LIST-TO-TREE (LT)
Transformation Rules

ACTION	RULE
REPLACE	(A)(B):=SYN(A;B);

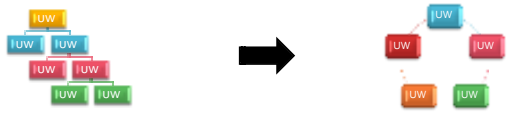
TREE-TO-TREE (TT)
Transformation Rules



ACTION	RULE
ADD RELATION	SYN(A;B):=+SYN(C;D);
DELETE RELATION	SYN(A;B):=-SYN(A;B);
REPLACE RELATION	SYN(A;B):=SYN(C;D);
MERGE RELATION	SYN(A;B)SYN(C;D):=SYN(E;F);
DIVIDE RELATION	SYN(A;B):=SYN(C;D)SYN(E;F);

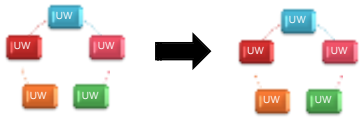
ACTION	RULE
ADD NODE	SYN(A;B):=SYN(A;B;C);
DELETE NODE	SYN(A;B):=SYN(A);

TREE-TO-NETWORK (TN)
Transformation Rules



ACTION	RULE
REPLACE	SYN(C;D):=SEM(A;B);:

NETWORK-TO-NETWORK (NN)
Transformation Rules



ACTION	RULE
ADD RELATION	SEM(A;B):=+SEM(C;D);
DELETE RELATION	SEM(A;B):=-SEM(A,B);
REPLACE RELATION	SEM(A;B):=SEM(C;D);
MERGE RELATION	SEM(A;B)SEM(C;D):=SEM(E;F);
DIVIDE RELATION	SEM(A;B):=SEM(C;D)SEM(E;F);

General Properties of
Transformation Rules

PRIORITY
Rules are applied serially, according to the order defined in the grammar. The first rule will be the first to be applied, the second will the second, and so on.

RECURSIVENESS
Rules are applied recursively as long as their conditions are true.

COMPREHENSIVENESS
Grammars are applied comprehensively as long as there is at least one applicable rule.

ACTION
The rules may add or delete values to the source and the target nodes, but only in the right side items:

agt(a;b):=agt(+c;);
agt(a;b):=agt(;-b);

General Properties of
Transformation Rules

CONSERVATION
Rules affect only the information clearly specified. No relation, node or feature is deleted unless explicitly informed.

For instance, in the examples below, the source node of the "agt" relation preserves, in all cases, the value "a". The only change concerns the feature "c", which is added to the source node of the "agt" in the first two cases; and the feature "b", which is deleted from the target node in the third case.

agt(a;b):=agt(c;);
agt(a;b):=agt(+c;);
agt(a;b):=agt(;-b);

In any case, the ADD and DELETE rules (i.e., when the right side starts with "+"or "-") **preserve** the items in the left side, except for the explicitly deleted ones:

General Properties of
Transformation Rules

SCOPE
The REPLACE, MERGE and DIVIDE rules affect only their designated scopes.

NN may only replace, merge or divide semantic relations; TT may only replace, merge or divide syntactic relations; and LL may only replace, merge or divide list nodes. All other information is preserved, unless explicitly informed.

INPUT: agt(a;b) cob(a;c)
RULE: cob(;):=obj(;);
OUTPUT: agt(a;b) obj(a;c)

INPUT: agt(a;b) cob(a;c)
RULE: cob(a;):=obj(-a,+d;);
OUTPUT: agt(a;b) obj(d;c)

General Properties of
Transformation Rules

CONJUNCTION
Both the left and the right side of the rule may have as many items as necessary. The items must be juxtaposition.
SEM(A;B)SEM(C;D)SEM(E;F):=SEM(G;H)SEM(I;J)SEM(K;L);

DISJUNCTION
The left side of the rules may bring disjuncts, but not the right side.

{SEM(A;B)|SEM(C;D)}^SEM(E;F):=+SEM(E;F);
SEM(A;B){SEM(C;D)|SEM(E;F)}:=-SEM(A;B);
agt(VER,{Vo1|Vo2};NOU,^SNG}):=;

General Properties of
Transformation Rules

COMMUTATIVITY
Inside the same side of the rule, the order of the factors does not affect the end result, except for list-processing rules (LL, LT and TL).

SEM(A;B):=SEM(C;D)SEM(E;F); = SEM(A;B):= SEM(E;F)SEM(C;D);
SYN(A;B):=SYN(C;D)SYN(E;F); = SYN(A;B):= SYN(E;F)SYN(C;D);

But:
(A):=(B)(C); ≠ (A):=(C)(B);
SYN(A;B):=(C)(D); ≠ SYN(A;B):=(D)(C);
(C)(D):=SYN(A;B); ≠ (D)(C):=SYN(A;B);

Additionally, the order of the features inside a relation does not affect the end result, but the order of the nodes is non-commutative.
SEM(VER,TRA ; NOU,MCL) = SEM(TRA,VER ; MCL,NOU)
But:
SEM(VER,TRA ; NOU,MCL) ≠ SEM(NOU,MCL ; VER,TRA)

General Properties of Transformation Rules

INDEXATION

- Indexes (%) are used for co-indexing nodes, attributes and values inside and between the left and the right side of transformation rules.
  - X(%a;)Y(%a;)
  - X(%a;%b):=Y(%b;%a);
- Any co-indexation is made by the use of indexes and not by the repetition of features.
  - **X(A;)Y(A;)** is different from **X(%a;)Y(%a;)**.
- Indexes are made of any sequence of alphanumeric characters and underscore (numbers are used for default indexation and must be avoided)

General Properties of Transformation Rules

INDEXATION

- Default indexation
  - If omitted, indexes are assigned by default
  - In default indexation, left-side nodes are automatically co-indexed with right-side nodes **if and only if** their position and number are the same:
    - X(A;B):=Y(C;D); is the same as X(**%01**,A;**%02**,B):=Y(**%01**,C;**%02**,D);
  - Non-co-indexed nodes in the right side means ADDITION, whereas left-side nodes that are not referred to in the right side means DELETION
    - X(%a;%b):=Y(%a;X;%b); is the same as X(%a;%b):=Y(%a;**%02**,X,,%b);
  - Indexes may also be used to transfer attribute values expressed in the format ATTRIBUTE=VALUE
    - X(A,%a,ATT1=VAL1;B,%b):=X(%a;%b,ATT1=%a);

English Analysis Grammar (1)

- SOURCE (eng)
  - asdfghij
- TARGET (UNL)
  - "asfghij"
- RULES
  - No rule is necessary

English Analysis Grammar (2)

- SOURCE (eng)
  - book
- TARGET (UNL)
  - book
- RULES
  - No rule is necessary

English Analysis Grammar (3)

- SOURCE (eng)
  - a book
- TARGET (UNL)
  - book.@indef
- RULES
  - DELETE THE BLANK SPACE
    - (" "):=;
  - PROCESS THE ARTICLE
    - ("a",%x)(N,%y):=(%y,+@indef); **OR**
    - (ART,@indef,%x) (N,%y):=(%y,+@indef); **OR**
    - (D,att,%x) (N,%y):=(%y,+att=%x);

English Analysis Grammar (4)

- SOURCE (eng)
  - the book
- TARGET (UNL)
  - book.@def
- RULES
  - DELETE THE BLANK SPACE
    - (" "):=;
  - PROCESS THE ARTICLE
    - ("the",%x)(N,%y):=(%y,+@def); **OR**
    - (ART,@def,%x) (N,%y):=(%y,+@def); **OR**
    - (D,att,%x) (N,%y):=(%y,+att=%x);

English Analysis Grammar (5)

- SOURCE (eng)
  - that book
- TARGET (UNL)
  - book.@distal
- RULES
  - DELETE THE BLANK SPACE
    - (" "):=;
  - PROCESS THE DEMONSTRATIVE
    - ("that",%x)(N,%y):=(%y,+@distal); **OR**
    - (DEM,@distal,%x) (N,%y):=(%y,+@distal); **OR**
    - (D,att,%x) (N,%y):=(%y,+att=%x);

English Analysis Grammar (6)

- SOURCE (eng)
  - this book
- TARGET (UNL)
  - book.@proximal
- RULES
  - DELETE THE BLANK SPACE
    - (" "):=;
  - PROCESS THE DEMONSTRATIVE
    - ("this",%x)(N,%y):=(%y,+@proximal); **OR**
    - (DEM,@proximal,%x) (N,%y):=(%y,+@proximal); **OR**
    - (D,att,%x) (N,%y):=(%y,+att=%x);

English Analysis Grammar (7)

- SOURCE (eng)
  - all books
- TARGET (UNL)
  - book.@all
- RULES
  - DELETE THE BLANK SPACE
    - (" "):=;
  - PROCESS THE QUANTIFIER
    - ("all",%x)(N,%y):=(%y,+@all); **OR**
    - (QUA,@all,%x) (N,%y):=(%y,+@all); **OR**
    - (D,att,%x) (N,%y):=(%y,+att=%x);

English Analysis Grammar (11)

- SOURCE (eng)
  - very beautiful
- TARGET (UNL)
  - beautiful.@plus
- RULES
  - DELETE THE BLANK SPACE
    - (" "):=;
  - PROCESS THE INTENSIFIER
    - ("very",%x)(J,%y):=(%y,+@plus); **OR**
    - (QUA,@plus,%x) (J,%y):=(%y,+@plus); **OR**
    - (SAV,att,%x) (J,%y):=(%y,+att=%x);

English Analysis Grammar (13)

- SOURCE (eng)
  - The books
- TARGET (UNL)
  - book.@def.@pl
- RULES
  - DELETE THE BLANK SPACE
    - (" "):=;
  - PROCESS THE ARTICLE
    - (D,att,%x) (N,%y):=(%y,+att=%x);
  - PROCESS THE NUMBER
    - (N,PLR,^@pl,%x):=(+@pl,%x);

TASK #5a

- Goal
  - To create a NL-UNL grammar to generate the corpus from your native language into UNL from sentences 1 to 17