Meaning construction

Philipp Cimiano · Christina Unger
Semantic Computing Group
CITEC, Bielefeld University
Today

Application domain

ONTOLOGY

LEXICON

Grammar
Generation

Reasoner

Ontology-based
interpretation

QA System

Grammar

class

closed
temporal

domain

...
Today

- **Grammars**: syntactic and semantic representations
- **Aligning** these representations to an ontology
- **Interpretation**: from a natural language string to an ontology-specific meaning representation
Grammars: Form and meaning

Grammars are explicit descriptions of the rules of a language.

- **Levels of description:**
  - Phonology and morphology explore sounds and how they form word parts and words.
  - **Syntax** studies how words are combined into phrases and sentences.
  - **Semantics** investigates the meanings of basic expressions, and how they are combined into meanings of more complex expressions.
The choice of grammar formalism

**Wanted:** A grammar formalism that proves well-suited for ontology-based interpretation of natural language.

- **Syntax:** Lexicalized Tree Adjoining Grammars (LTAG)
- **Semantics:** Dependency-based Underspecified Discourse Representation Structures (DUDES)

...mainly because of their flexibility w.r.t. atomic building blocks and composition of elements.
Outline

Syntactic representations
  LTAG
  Aligning syntactic representations to an ontology

Semantic representations
  DUDES

Pairing syntactic and semantic representations

Grammar engineering: Example
Outline

Syntactic representations
   LTAG
   Aligning syntactic representations to an ontology

Semantic representations
   DUDES

Pairing syntactic and semantic representations

Grammar engineering: Example
Outline

Syntactic representations
  LTAG
    Aligning syntactic representations to an ontology

Semantic representations
  DUDES

Pairing syntactic and semantic representations

Grammar engineering: Example
Tree Adjoining Grammars (TAG) is a mildly context-sensitive grammar formalism that builds on trees as representations of syntactic structure, even as basic building blocks.

Two basic ingredients:
- elementary trees (initial and auxiliary trees)
- structure-building operations that expand and combine trees (substitution and adjunction)
Trees

- Leaf nodes are strings.
- Branching nodes are labelled with syntactic categories.

**Example:**

```
S
  /   \
/     \V
DP   VP
  \   /DP
   \ /DP
    \ /NP
     \NP
```

Uruguay V DP

Uruguay won DET NP

the game
Locality domains

**TAG's fundamental hypothesis (Frank 2002)**
Every syntactic dependency is expressed locally within a single elementary tree.

**Constraint on elementary trees (Frank 2002)**
Every elementary tree comprises the extended projection of a single lexical head – nothing more and nothing less.

- keep elementary trees as minimal as possible
- at the same time allow them to be as big as necessary to contain all elements that directly depend on the lexical head
Examples

Syntactic representations

Uruguay

subject

won

object
Examples

DP

Uruguay

S

VP

V

DP

won
A TAG grammar is lexicalized if each elementary tree is associated with at least one lexical element, i.e. contains at least one leaf node labelled with a terminal symbol (the anchor).
**Combining trees: Substitution**

**Substitution** applies to:

- a tree with a leaf node $C \downarrow$ (where $C$ is any syntactic category)
- an initial tree with $C$ as root node

The latter is substituted for the leaf node $C \downarrow$.

**Example:**

```
     A
  /   \
B     C
     \  
      D  E
```
Combining trees

Syntactic representations
Combining trees

Syntactic representations

Univ of Uruguay

VP

won

DP

V

DP
Combining trees

[Diagram of a tree structure showing DP, DET, and NP nodes with the words 'the' and 'game']
Combining trees

```
DP
  DET NP
    the  game
```
Combining trees

Syntactic representations
Combining trees

S

DP
Uruguay

VP
V
won

DP
DET
NP
the
game

Syntactic representations

LTAG
Adjectives

Specify an elementary tree for hardest.
Adverbs

Specify an elementary tree for easily.

Syntactic representations
Combining trees: Adjunction

Adjunction applies to

- an auxiliary tree with a footnote \( C^* \) and a root \( C \)
- a tree with some node \( C \)

The node \( C \) is replaced by the auxiliary tree, where the foot node is replaced by the subtree rooted by \( C \).

Example:

```
A
 /   \
B     C
    /   \
   D     E
```

```
C
 /   \
C^*  D
```

Example

S
  /\    /
DP   VP
  |    /\       /\ 
  |    V  DP   VP
  |    |   |    | 
  won|   |   easily

Syntactic representations

LTAG
Example

Syntactic representations

LTAG
Example

```
S
  └── DP
  └── VP
    └──ADV
        └── easily
    └── VP
        └── DP
            └── V
                └── won
```
Outline

Syntactic representations
- LTAG
  Aligning syntactic representations to an ontology

Semantic representations
- DUDES

Pairing syntactic and semantic representations

Grammar engineering: Example
Ontology-based LTAG

**Example:** The relation `soccer:atMinute` can be verbalised as in the `[num] minute`. But: Which meanings to assign to these elementary trees?
**Example:** The relation `soccer:atMinute` can be verbalised as *in the [num] minute.*

**Elementary trees:**

```
  VP
   VP* PP
   P    DP
       in

dp
   DET NP
       the

np
   NUM NP*
       [num]

np
   minute
```
Ontology-based LTAG

**Example:** The relation `soccer:atMinute` can be verbalised as `in the [num] minute`.

**Elementary trees:**

```
   VP  
  /   
 VP* PP  
 |   |  
 P   DP  
 |   |  
 in
```

```
   DP  
  /   
 DET NP  
 |   |  
 the
```

```
   NP  
  /   
 NUM NP*  
 |   |  
 [num] minute
```

**But:** Which meanings to assign to these elementary trees?
Ontology-based LTAG

Elementary tree:

```
VP
  VP
  VP*
  PP
  P
  in
  DET
  the
  NUM
  NP
  minute
```
Ontology-based LTAG

That is, we need to assume more complex elementary trees.

**Rule of thumb**
One ontology concept, one elementary tree.

**Ontology-based elementary trees**
An elementary tree spans the extended projections of all lexical items that are required to verbalise a single ontology concept.
LTAG derivations

- Derived tree:
  result of carrying out substitutions and adjunctions
  (syntactic output)

- Derivation tree:
  record of the history of how elementary trees are put together
  (usually input to semantics)
Compositional semantics:

- Associate each elementary tree with a formal meaning representation that connects it to the ontology concept it verbalizes.
- Associate each syntactic operation for combining trees with a semantic operation for combining semantic representations.
Outline

Syntactic representations
  LTAG
  Aligning syntactic representations to an ontology

Semantic representations
  DUDES

Pairing syntactic and semantic representations

Grammar engineering: Example
Outline

Syntactic representations
  LTAG
  Aligning syntactic representations to an ontology

Semantic representations
  DUDES

Pairing syntactic and semantic representations

Grammar engineering: Example
DUDES are extensions of structures from Discourse Representation Theory (DRT).
Idea behind DRT: Interpretation in context

- Each sentence of a discourse is interpreted in the context of the preceding sentences.
- The context is updated with the contribution of the sentence, yielding a new context in which subsequent sentences are interpreted.
- This update often involves connecting elements of the sentence with elements from the context (e.g. pronouns with their antecedents).

Content and context are represented as Discourse Representation Structures (DRSs).
A DRS consists of two parts:

- a set of reference markers (or: discourse referents) representing the entities that a discourse is about
- a set of conditions

**Example:** A player from Uruguay scored a goal.

<table>
<thead>
<tr>
<th>x, y</th>
</tr>
</thead>
<tbody>
<tr>
<td>player(x)</td>
</tr>
<tr>
<td>from(x, uruguay)</td>
</tr>
<tr>
<td>goal(y)</td>
</tr>
<tr>
<td>score(x, y)</td>
</tr>
</tbody>
</table>
Ontology-based DRSs

**Example:** A player from Uruguay scored a goal.

<table>
<thead>
<tr>
<th>x, y, r</th>
</tr>
</thead>
<tbody>
<tr>
<td>soccer:Goal(y)</td>
</tr>
<tr>
<td>soccer:byPlayer(y, x)</td>
</tr>
<tr>
<td>soccer:role(x, r)</td>
</tr>
<tr>
<td>soccer:PlayerRole(r)</td>
</tr>
<tr>
<td>soccer:team(r, soccer:Uruguay)</td>
</tr>
</tbody>
</table>
Ontology-based DRSs

Every non-logical constant (predicate or individual constant) corresponds to an ontology concept or entity.

- **Classes**: unary predicates
- **Relations**: n-nary predicates
- **Entities**: individual constants
Outline

Syntactic representations
  LTAG
  Aligning syntactic representations to an ontology

Semantic representations
  DUDES

Pairing syntactic and semantic representations

Grammar engineering: Example
Example

Uruguay won against Argentina.

```
S
  DP
    Uruguay
  VP
    V
      won
    PP
      P
        against
      DP
        Argentina
```

<table>
<thead>
<tr>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>soccer : Match(x)</td>
</tr>
<tr>
<td>soccer : winner(x, soccer : Uruguay)</td>
</tr>
<tr>
<td>soccer : loser(x, soccer : Argentina)</td>
</tr>
</tbody>
</table>

Pairing syntactic and semantic representations
Elementary trees with meanings

DP
Uruguay

S
DP
VP
V
won
PP
against

DP
Argentina

u
u = soccer : Uruguay

z
soccer:Match(z)
soccer:winner(z, x)
soccer:loser(z, y)

v
v = soccer : Argentina
Elementary trees with meanings

DP
Uruguay

S

DP₁ ↓

VP

V
won

PP
against

DP₂ ↓

z
soccer:Match(z)
soccer:winner(z, x)
soccer:loser(z, y)
(DP₁, x), (DP₂, y)

u
u = soccer : Uruguay

v

v = soccer : Argentina
Combining meanings

If a tree \( T \) with root \( C \) is substituted in another tree with a substitution node \( C \downarrow \) (i.e. the meaning of which is a DUDES with a selection pair \((C, v)\)), then the meanings of both trees are combined by

- taking the union of the universes, condition lists and selection pairs (except for the \((C, v)\)) of both DUDES
- unifying the selection variable \( v \) and the main variable of the semantic representation of the argument tree \( T \)
Example

Pairing syntactic and semantic representations:

- **S**
  - **DP<sub>1</sub>**
    - **DP**
      - **Uruguay**
  - **VP**
    - **V**
      - **won**
    - **PP**
      - **against**
  - **PP**
    - **DP<sub>2</sub>**

**v**
- **v = soccer : Uruguay**

**z**
- soccer:Match(z)
- soccer:winner(z, x)
- soccer:loser(z, y)
- (DP<sub>1</sub>, x), (DP<sub>2</sub>, y)
Example

Pairing syntactic and semantic representations

- **S**: Sentence
- **DP**: Determiner Phrase
- **VP**: Verb Phrase
- **PP**: Prepositional Phrase
- **V**: Verb
- **P**: Preposition

**Tree Structure**:

- **S**
  - **DP**
    - **DP1**
  - **VP**
    - **V**: won
    - **PP**
      - **P**: against
      - **DP2**

**Semantic Representations**:

- **x**
  - **x = soccer : Uruguay**

- **z**
  - **soccer:Match(z)**
  - **soccer:winner(z, x)**
  - **soccer:loser(z, y)**
  - **(DP1, x), (DP2, y)**
Example

Pairing syntactic and semantic representations

```
S
  DP
    Uruguay
  VP
    V
      won
    PP
      against
```

<table>
<thead>
<tr>
<th>z, x</th>
</tr>
</thead>
<tbody>
<tr>
<td>x = soccer : Uruguay</td>
</tr>
<tr>
<td>soccer:Match(z)</td>
</tr>
<tr>
<td>soccer:winner(z, x)</td>
</tr>
<tr>
<td>soccer:loser(z, y)</td>
</tr>
<tr>
<td>(DP₂, y)</td>
</tr>
</tbody>
</table>
Example

Pairing syntactic and semantic representations

```
S
   /\  \
  DP  VP
 /\   /\  \
Uruguay  V  PP
   \   \       
     won  P       DP
          \  /       
          DP_2  against
          \       
          Argentina
```

<table>
<thead>
<tr>
<th>z, x</th>
</tr>
</thead>
<tbody>
<tr>
<td>x = soccer : Uruguay</td>
</tr>
<tr>
<td>soccer:Match(z)</td>
</tr>
<tr>
<td>soccer:winner(z, x)</td>
</tr>
<tr>
<td>soccer:loser(z, y)</td>
</tr>
<tr>
<td>(DP_2, y)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>v = soccer : Argentina</td>
</tr>
</tbody>
</table>

Example

Pairing syntactic and semantic representations
Example

Pairing syntactic and semantic representations

\[
\text{DP} \quad \text{VP}
\]

\[
\text{Uruguay} \quad \text{V} \quad \text{PP}
\]

\[
\text{won} \quad \text{P} \quad \text{DP}
\]

\[
\text{against} \quad \text{Argentina}
\]

\[
z, x, y
\]

\[
y = \text{soccer: Argentina}
\]

\[
x = \text{soccer: Uruguay}
\]

\[
\text{soccer:Match}(z)
\]

\[
\text{soccer:winner}(z, x)
\]

\[
\text{soccer:loser}(z, y)
\]
DUDES additionally contain information regarding:

- the main variable
- semantic types (the usual: $\tau ::= e \mid t \mid (\tau \to \tau)$)
- labels of sub-DRSs
- relative quantifier scope (underspecified)
Outline

Syntactic representations
  LTAG
  Aligning syntactic representations to an ontology

Semantic representations
  DUDES

Pairing syntactic and semantic representations

Grammar engineering: Example
Which syntactic and semantic representations would we need to cover the following sentences?

- Franck Ribery scored.
- Franck Ribery scored a goal.
- Franck Ribery scored a penalty kick.
Grammar engineering: Example

Tomorrow

Application domain

Ontology

LEXICON

Grammar Generation

Reasoner

QA System

Ontology-based interpretation

closed class

temporal

...