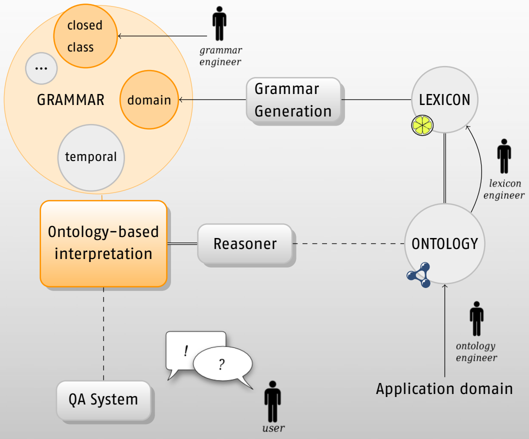
August 14, 2012

Meaning construction

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

Today


closed
class
...
GRAMMAR
domain
temporal
grammar engineer
Grammar Generation
LEXICON
lexicon engineer
Ontology-based interpretation
Reasoner
ONTOLOGY
ontology engineer
!
?
user
QA System
Application 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

Universität Bielefeld

Grammars: Form and meaning

Grammars are explicit descriptions of the rules of a language.

- Levels of description:
  - Phonoloy 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.

CITEC

Universität Bielefeld

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.

CITEC

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

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.

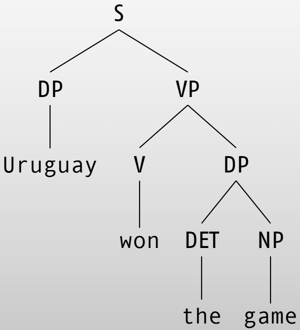
Two basic ingredients:

- 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:



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.

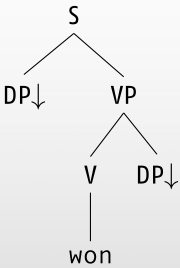
- keep elementary trees as minimal as possible
- at the same time allow them to be as big as necessary to contain all elements that directly depend on the lexical head

Examples

DP
Uruguay

S
DP VP
subject V DP
won object

Examples



Universität Bielefeld

Lexicalized TAG

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

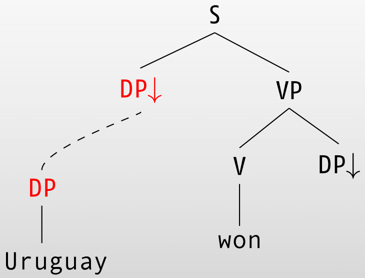
Substitution applies to

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

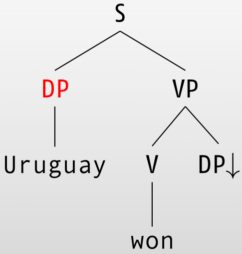
The latter is substituted for the leaf node C ↓.

Example:

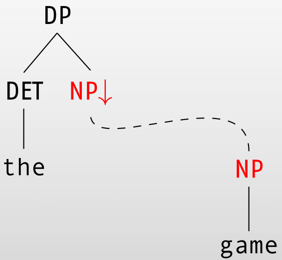
Combining trees



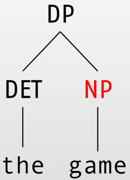
Combining trees



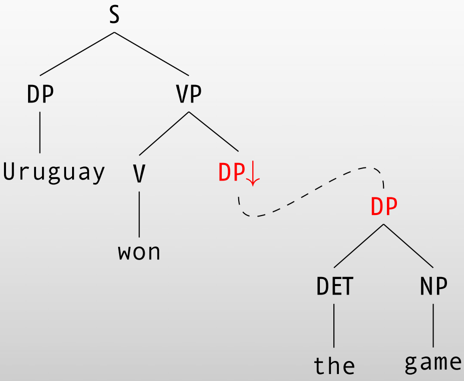
Combining trees



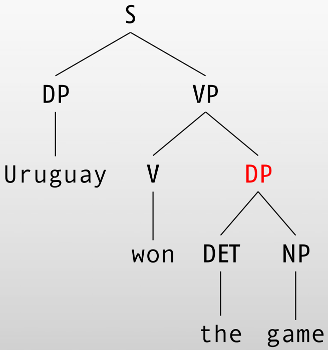
Combining trees



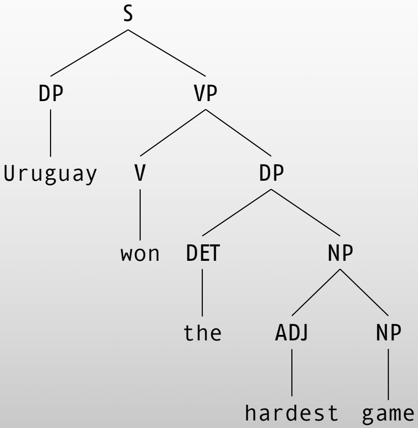
Combining trees



Combining trees

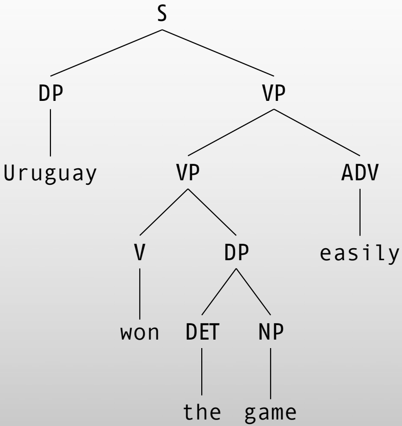


Adjectives



Specify an elementary tree for `hardest`.

Adverbs



Specify an elementary tree for easily.

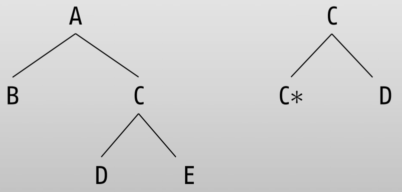
Combining trees: Adjunction

Adjunction applies to

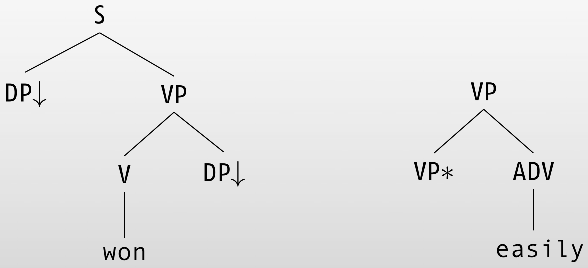
- an auxiliary tree with a footnote C∗ and a root C
- a tree with some node C

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

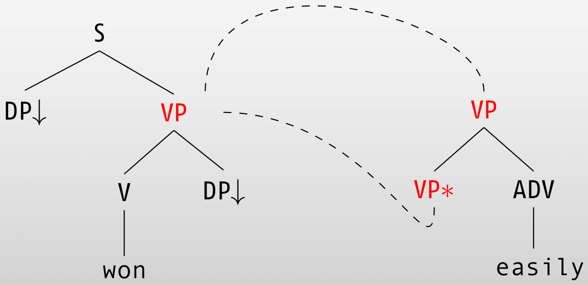
Example:



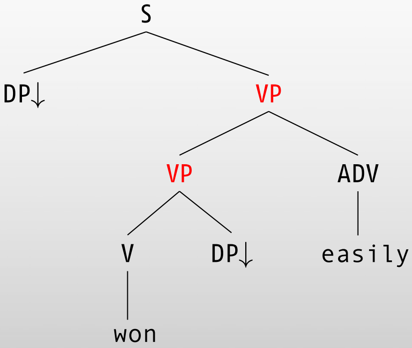
Example



Example



Example



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

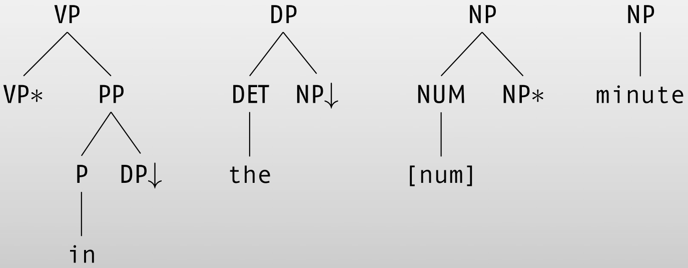
`in the [num] minute`.

Ontology-based LTAG

Example: The relation `soccer:atMinute` can be verbalised as

`in the [num] minute`.

Elementary trees:

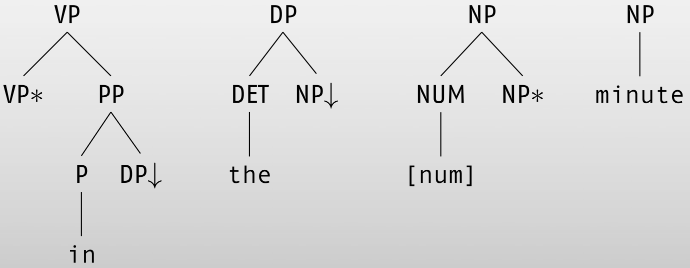


Ontology-based LTAG

Example: The relation `soccer:atMinute` can be verbalised as

`in the [num] minute`.

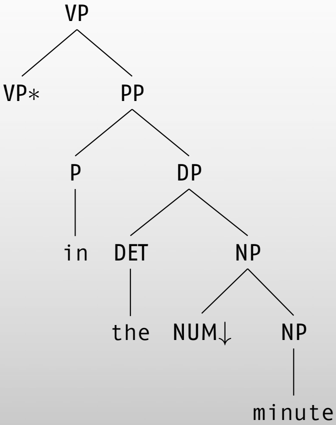
Elementary trees:



But: Which meanings to assign to these elementary trees?

Ontology-based LTAG

Elementary tree:



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)

Next...

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

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).

Discourse representation structures

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.`

x, y
player(x) from(x, uruguay) goal(y) score(x, y)

Universität Bielefeld

Ontology-based DRSs

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

x, y, r
soccer:Goal(y) soccer:byPlayer(y, x) soccer:role(x, r) soccer:PlayerRole(r) soccer:team(r, soccer:Uruguay)

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

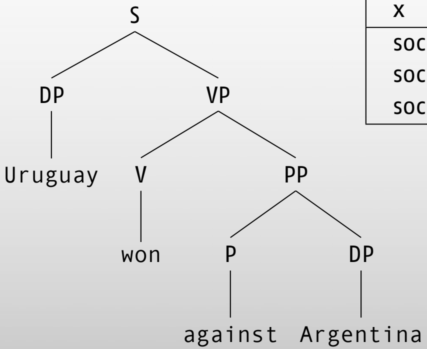
- DUDES

Pairing syntactic and semantic representations

Grammar engineering: Example

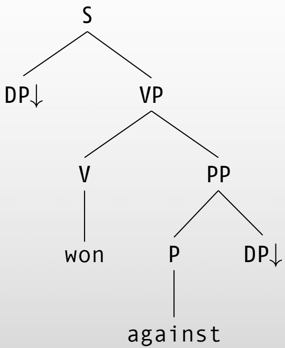
Example

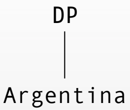
`Uruguay won against Argentina.`



x
soccer : Match(x) soccer : winner(x, soccer : Uruguay) soccer : loser(x, soccer : Argentina)

Elementary trees with meanings



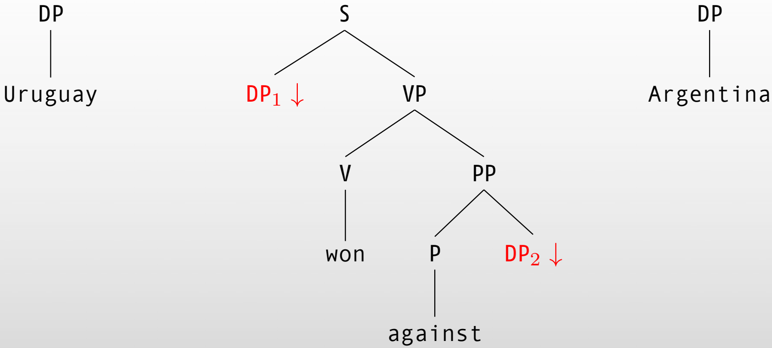


u
u = soccer : Uruguay

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

v
v = soccer : Argentina

Elementary trees with meanings



u
u = soccer : Uruguay

z
soccer:Match(z) soccer:winner(z, x) soccer:loser(z, y)
$(\mathrm{DP}_1, x), (\mathrm{DP}_2, y)$

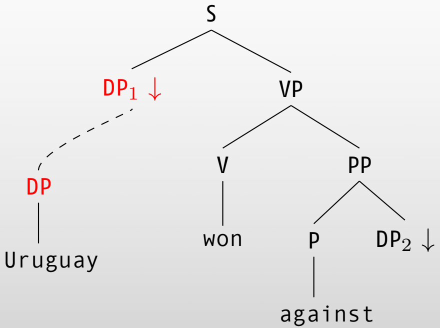
v
v = soccer : Argentina

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

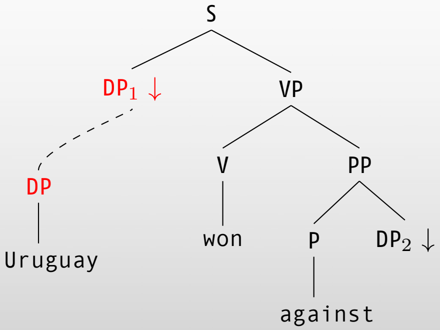
Example



v
v = soccer : Uruguay

z
soccer:Match(z) soccer:winner(z, x) soccer:loser(z, y)
(DP_1, x), (DP_2, y)

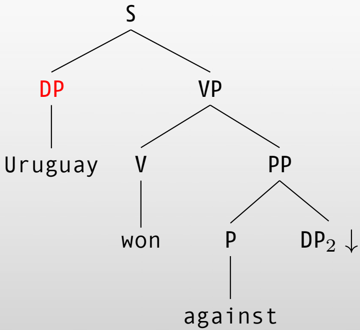
Example



x
x = soccer : Uruguay

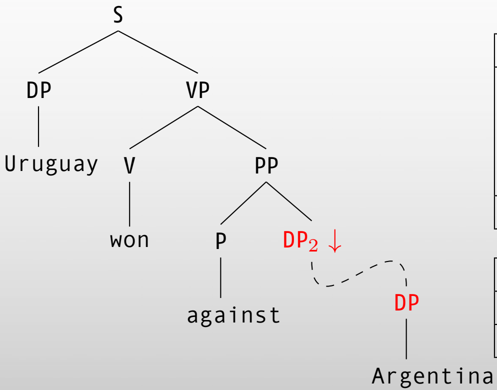
z
soccer:Match(z) soccer:winner(z, x) soccer:loser(z, y)
$(DP_1, x), (DP_2, y)$

Example



z, x
$x =$ soccer : Uruguay soccer:Match(z) soccer:winner(z, x) soccer:loser(z, y)
(DP_2, y)

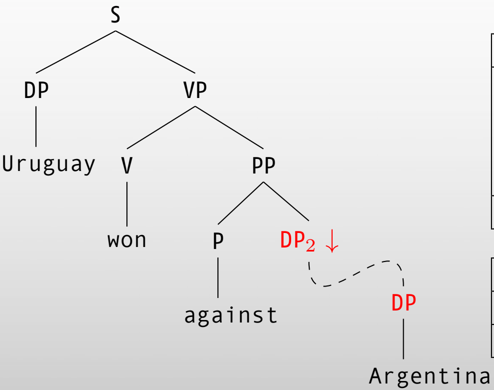
Example



z, x
x = soccer : Uruguay soccer:Match(z) soccer:winner(z, x) soccer:loser(z, y)
(DP_2, y)

v
v = soccer : Argentina

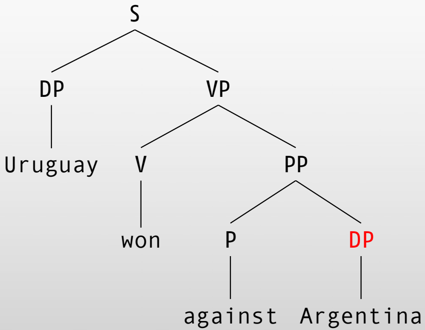
Example



z, x
$x = $ soccer : Uruguay soccer:Match(z) soccer:winner(z, x) soccer:loser(z, y)
(DP_2, y)

y
$y = $ soccer : Argentina

Example



z, x, y
y = soccer : Argentina x = soccer : Uruguay soccer:Match(z) soccer:winner(z, x) soccer:loser(z, y)

DUDES

DUDES additionally contain information regarding:

- the main variable
- semantic types (the usual: $\tau \ ::= \ \mathsf{e} \mid \mathsf{t} \mid (\tau \rightarrow \tau)$)
- 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

Examples

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.

Universität Bielefeld

Tomorrow