Universität Bielefeld

August 13, 2012

Introduction

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

CITEC

Course website:

http://greententacle.techfak.uni-bielefeld.de/~cunger/esslli2012/

Outline

Natural language interpretation

Ontologies

Course overview

Outline

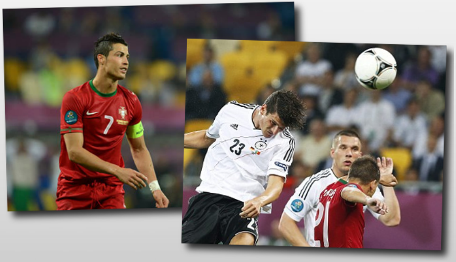
Natural language interpretation

Ontologies

Course overview

Natural language interpretation

- The 90th minute header of Gómez cancelled out the early opener by Ronaldo.



Boxer in action

- The 90th minute header of Gómez cancelled out the early opener by Ronaldo.

x0,x1,x2,x3,x4
named(x0,gomez,loc)
minute(x1)
90th(x2)
header(x2)
nn(x1,x2)
of(x2,x0)
named(x3,ronaldo,loc)
early(x4)
opener(x4)
by(x4,x3)

;

x5
cancel(x5)
out(x5)
event(x5)
agent(x5,x2)
patient(x5,x4)

Natural language understanding

- What exactly is the task of `understanding' natural language?
- When would we say that a computer program `understands' language?
- How would you check?

Towards a definition of natural language understanding

Definition: Natural language understanding

A computer program is said to understand sentence s if the situations in which the sentence is true for the machine correspond to the (mental) situations in which the sentence is true for a human.

Natural language understanding

- The 90th minute header of Gómez cancelled out the early opener by Ronaldo.

Model-theoretic semantics

- Model-theoretic semantics defines structures that represent different states the world can be in (models) and specifies when (a formal representation of) a natural language expression is true with respect to such a model.
- An expression e' is said to follow from expression e in case the models $M(e')$ that satisfy e' are a subset of the models that satisfy e, i.e. $M(e') \subseteq M(e)$.

A simple example

- The Netherlands lost the soccer match against Denmark.

The representation of this sentence in some first-order like language might be as follows:

- $\exists e, d, n, t\,.\, \text{soccer\_match}(e) \wedge \text{lose\_against}(e, n, d) \wedge \text{name}(n) = \text{`The Netherlands'} \wedge \text{name}(d) = \text{`Denmark'} \wedge \text{happens}(e, t) \wedge \text{end}(t) < \text{now}$

A simple example

- The Netherlands lost the soccer match against Denmark.

Now, let us ask a few simple questions:

- Who was the winner in the soccer match between the Netherlands and Denmark?
- Are the Netherlands and Denmark soccer teams?

Universität Bielefeld

Another example

- Rooney scored a penalty kick in a 3–1 home victory against Switzerland.

As a first approximation of this, we might represent the meaning of this sentence as follows:

- $\exists$e, p, s, r, v . name(r) = \`Rooney' $\wedge$ penalty_kick(p) $\wedge$ scored(e, r, p) $\wedge$ in(e, v) $\wedge$ 3_1_home_victory(v) $\wedge$ against(v, s) $\wedge$ name(s) = \`Switzerland'

CITEC

Another example

- Rooney scored a penalty kick in a 3–1 home victory against Switzerland.

As humans, we have no trouble answering the questions below:

- Who lost the game?
- Who won the game?
- Where did the game take place?
- How many goals were scored in the game?

Another example

- Rooney scored a penalty kick in a 3–1 home victory against Switzerland.

A more elaborate representation:

- $\exists$e, p, ft, s, r, g, t . soccer_match(e)$\wedge$ team(e, ft) $\wedge$ team(e, s) $\wedge$ penalty_kick(p) $\wedge$ inMatch(p, e) $\wedge$ byPlayer(p, r) $\wedge$ name(r) $=$ `Rooney' $\wedge$ leadsTo(p, g) $\wedge$ Goal(g) $\wedge$ host(e, ft) $\wedge$ visitor(e, s) $\wedge$ final_score(e, ft) $= 3 \wedge$ final_score(e, s) $= 1 \wedge$ winner(e, ft) $\wedge$ name(s) $=$ `Switzerland' ($\wedge$ happens(e, t) $\wedge$ plays_for(r, ft, t$'$) $\wedge$ t $\subseteq$ t$'$)

Coming back to our example questions

- Who lost the game?
- Who won the game?
- Where did the game take place?
- How many goals were scored in the game?

In order to answer them, we rely on background knowledge, which we can capture in meaning postulates.

Universität Bielefeld

Meaning postulates

Meaning postulate 1:

There are exactly two teams per match.

$$\forall e\,.\,\text{soccer\_match}(e) \rightarrow \exists t_1, t_2\,.\,\text{team}(e, t_1) \wedge \text{team}(e, t_2) \wedge t_1 \neq t_2 \wedge \forall t\,.\,\text{team}(e, t) \rightarrow t = t_1 \vee t = t_2$$

Therefore Rooney needs to play for either Switzerland or the hosting team.

Meaning postulates

Meaning postulate 2:
There is only one winner per match.

$$\forall e, w \, . \, soccer\_match(e) \wedge winner(e, w) \rightarrow \forall y \, . \, winner(e, y) \rightarrow y = w$$

Universität Bielefeld

Meaning postulates

Meaning postulate 3:

If one team is the winner, then the other one is the loser.

$$\forall e, x, y \,.\; (soccer\_match(e) \wedge team(e,x) \wedge team(e,y) \wedge x \neq y \wedge winner(e,x)) \rightarrow loser(e,y)$$

Universität Bielefeld

Meaning postulates

Meaning postulate 4:

If one team is the loser, then the other team is the winner.

$$\forall e, x, y \,.\; (\text{soccer\_match}(e) \wedge \text{team}(e, x) \wedge \text{team}(e, y) \wedge x \neq y \wedge \text{loser}(e, x)) \rightarrow \text{winner}(e, y)$$

Meaning postulates

Meaning postulate 5:
A team is the winner if and only if its final score is higher than that of the other team.

$$\forall e, x, y\,.\, \text{soccer\_match}(e) \rightarrow (\text{winner}(e, x) \wedge \text{loser}(e, y)) \leftrightarrow (\text{final\_score}(e, x) > \text{final\_score}(e, y))$$

Universität Bielefeld

Meaning postulates

Meaning postulate 6:

The number of goals is the sum of the final scores of both teams.

$$\forall e, t_1, t_2 \,.\; (\mathrm{soccer\_match}(e) \wedge \mathrm{team}(e, t_1) \wedge \mathrm{team}(e, t_2) \wedge t_1 \neq t_2) \\ \rightarrow \mathrm{total\_goals}(e) = \mathrm{final\_score}(e, t_1) + \mathrm{final\_score}(e, t_2))$$

Meaning postulates

Meaning postulate 7:
The location of a match corresponds to the location of the hosting team.

$$\forall e, t, l \,.\; \text{soccer\_match}(e) \wedge \text{hostingTeam}(e, t) \wedge \text{location}(t, l) \rightarrow \text{location}(e, l)$$

Meaning postulates

- Such meaning postulates (or logical axioms) essentially encode the knowledge that we humans have about the world.
- Equipped with them, a computer program could draw the right inferences to answer the above questions.

Even more...

Meaning postulates are required to have a deep understanding of language.

How many meaning postulates do we need?

How many meaning postulates do we need?

Hypothesis

We need at least a million meaning postulates to develop a general (open domain) natural language understander.

A million?

Universität Bielefeld

How many meaning postulates do we need?

Hypothesis

We need at least a million meaning postulates to develop a general (open domain) natural language understander.

Assume:

- A vocabulary of 50.000 words (a lower bound for sure)
- 20 meaning postulates per word (a lower bound for sure as well)

This gives us a lower bound of a million meaning postulates.

(Surely, some meaning postulates might be reused per words, but still...)

Meaning postulates from heaven?

Possible strategies:

- Use no axioms at all
- Write meaning postulates by hand
- Develop a large general ontology about everything
- Learn meaning postulates automatically

Meaning postulates from heaven?

Possible strategies:

- Use no axioms at all → will not allow for deeper understanding
- Write meaning postulates by hand
- Develop a large general ontology about everything
- Learn meaning postulates automatically

Meaning postulates from heaven?

Possible strategies:

- Use no axioms at all → will not allow for deeper understanding
- Write meaning postulates by hand → not feasible
- Develop a large general ontology about everything
- Learn meaning postulates automatically

Meaning postulates from heaven?

Possible strategies:

- Use no axioms at all → will not allow for deeper understanding
- Write meaning postulates by hand → not feasible
- Develop a large general ontology about everything → not feasible
- Learn meaning postulates automatically

Meaning postulates from heaven?

Possible strategies:

- Use no axioms at all → will not allow for deeper understanding
- Write meaning postulates by hand → not feasible
- Develop a large general ontology about everything → not feasible
- Learn meaning postulates automatically → there has been some work, but methods are not perfect

Open domain fallacy

Problem:

Automatic natural language understanding without a restriction to a specific domain is not feasible.

Solution:

Restrict task to a given sub-world in the hope that all relevant concepts, axioms etc. can be modelled by a domain expert.
(Note: Domain here does not mean a genre.)

Ontology-based natural language interpretation

Definition: Ontology-based natural language interpretation

A computer program is said to interpret sentence s if it can map the sentence into a logical form with respect to the vocabulary defined by a given ontology $\mathcal{O}$.

Outline

Natural language interpretation

Ontologies

Course overview

Goals

Our goal when defining ontologies is that a system `understands' (or: `interprets') symbols and draws appropriate inferences that are in line with our own understanding of the real-word concepts that these symbols represent.

Example domain: Soccer

Concepts:

- soccer teams
- soccer players
- soccer matches
- ...

Symbols for representing these sets: `player`, `team`, `match`
(logical predicates of arity 1, making predications about which individuals belongs to each of these sets)

Example domain: Soccer

Concepts:

- a player belonging to a team
- a player participating in a match
- ...

Symbols for representing these relationships: `member`, `plays`
(logical predicates of arity 2, relating players with teams/matches)

Example domain: Soccer

Now for a machine, these are just arbitrary symbols. We as humans, however, have knowledge about them:

- A player cannot be a match or a team.
- A player cannot play in two different matches at the same time.
- If one of the teams in a match wins, then the other team loses.
- If one team has a higher score than the other team, then the former is the winner and the latter is the loser.
- ...

A computer lacks such knowledge and will have no trouble assuming that some entity can be both a team and a match, etc.

Introducing constraints

In order to prevent a computer program to use the symbols `in the wrong way', we want to constrain the use of symbols by stating things which are impossible:

- It is impossible that a match is at the same time a player, i.e. the set of players cannot share elements with the set of matches.
- It is impossible that a player plays at the same time in two matches.
- It is impossible that one team is loser and winner at the same time.
- It is impossible that there are two winners or two losers.
- ...

Ontologies

So essentially, an ontology introduces constraints on the way the vocabulary is used¸ in order to rule out unintended interpretations of this vocabulary.

We can use different logical languages to express such constraints.

- The Web Ontology Language (OWL) is the knowledge representation language standardized by the World Wide Web Consortium to formalize ontologies, based on description logics.

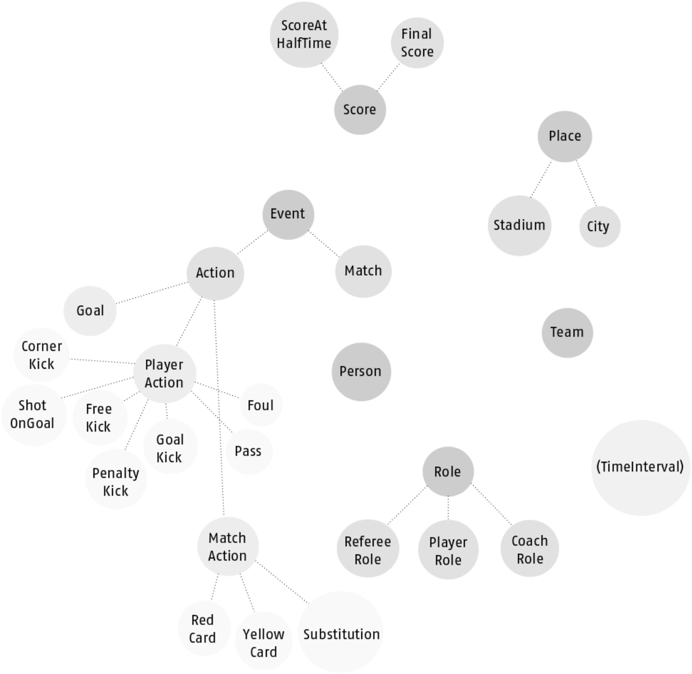
Ontologies

So essentially, an ontology introduces constraints on the way the vocabulary is used, in order to rule out unintended interpretations of this vocabulary.

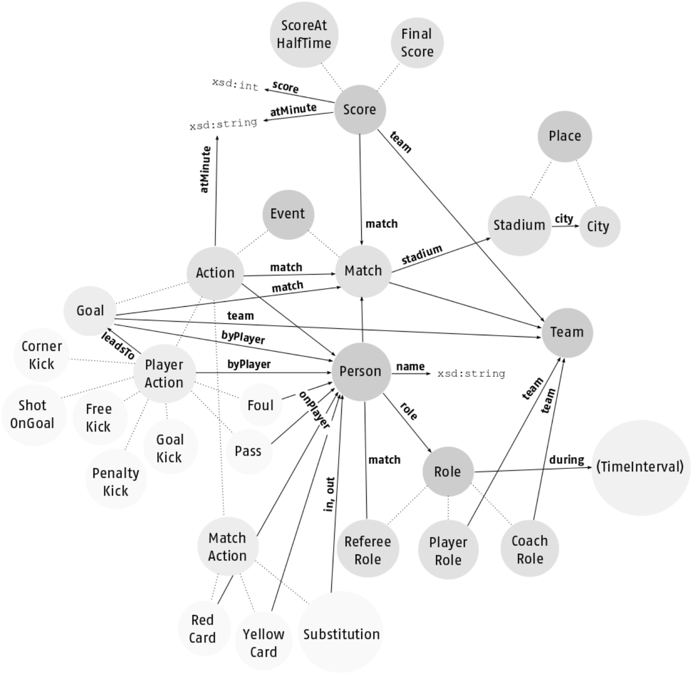
We can use different logical languages to express such constraints.

- The Web Ontology Language (OWL) is the knowledge representation language standardized by the World Wide Web Consortium to formalize ontologies, based on description logics.

Example: `soccer.owl`



ScoreAt HalfTime
Final Score
Score
Place
Stadium
City
Event
Action
Match
Goal
Team
Corner Kick
Player Action
Person
Shot OnGoal
Free Kick
Foul
Goal Kick
Pass
Penalty Kick
Role
(TimeInterval)
Match Action
Referee Role
Player Role
Coach Role
Red Card
Yellow Card
Substitution



ScoreAt HalfTime
Final Score
xsd:int
score
atMinute
xsd:string
Score
team
Place
atMinute
Event
match
Stadium
city
City
Action
match
Match
stadium
match
Goal
team
Team
byPlayer
leadsTo
Corner Kick
Player Action
byPlayer
Person
name
xsd:string
Shot OnGoal
Free Kick
Foul
onPlayer
role
team
team
Goal Kick
Pass
match
Role
during
(TimeInterval)
Penalty Kick
in, out
Match Action
Referee Role
Player Role
Coach Role
Red Card
Yellow Card
Substitution

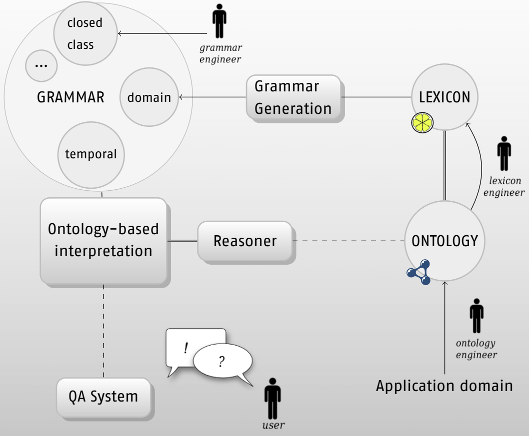
Outline

Natural language interpretation

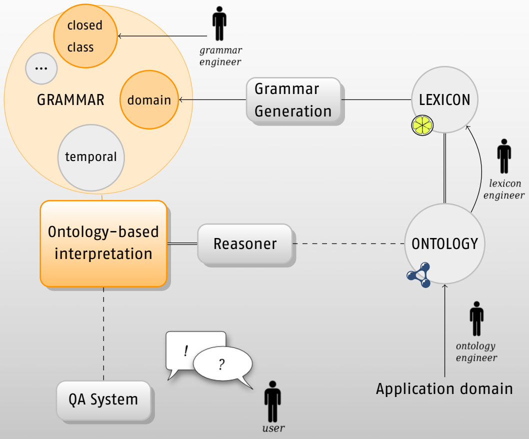
Ontologies

Course overview

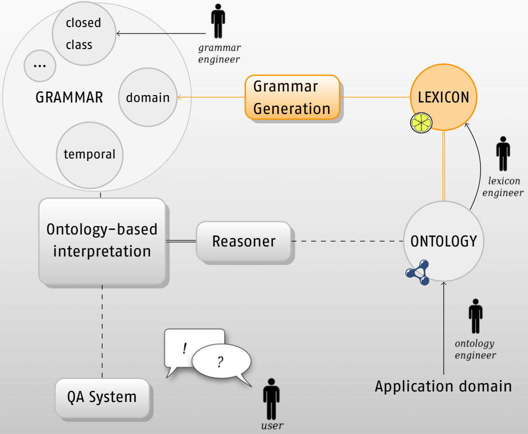
Overview


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

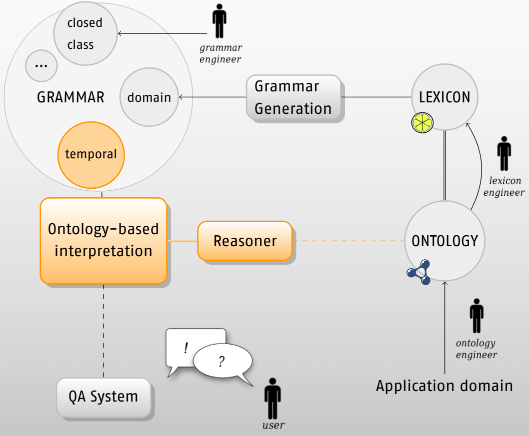
Tuesday


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

Wednesday


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

Thursday


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

Universität Bielefeld

Friday