Towards Robustness in Natural Language Understanding

Jordi Atserias i Batalla

Barcelona, 19 d'Abril

Introduction: is NLU that complex?

A child of five would understand this. Send someone to fetch a child of five.

Groucho Marxs 

Introduction: is NLU that complex?

An English professor wrote the following words on the blackboard:

“Woman without her man is nothing”

and directed his students to punctuate it correctly.

Introduction: is NLU that complex?

An English professor wrote the following words on the blackboard:

“Woman without her man is nothing”

and directed his students to punctuate it correctly.

The men wrote: **“Woman, without her man, is nothing.”**

5

Introduction: is NLU that complex?

An English professor wrote the following words on the blackboard:

"Woman without her man is nothing"

and directed his students to punctuate it correctly.

The men wrote: **"Woman, without her man, is nothing."**
The women wrote: **"Woman: Without her, man is nothing."**

6

1. Introduction
2. Knowledge Integration in PARDON
3. PARDON's Architecture
4. Experiments
5. Conclusions and Future Work

Outline

Outline

Introduction

There are several inherent difficulties in most NLU tasks:

1. The open and compositional nature of language (i.e. the difficulty of building complete repositories, or broad coverage grammars, etc).

Introduction

There are several inherent difficulties in most NLU tasks:

1. The open and compositional nature of language (i.e. the difficulty of building complete repositories, or broad coverage grammars, etc).
2. The inconsistencies (either coming from the models, the knowledge or the speaker).

Introduction

There are several inherent difficulties in most NLU tasks:

1. The open and compositional nature of language (i.e. the difficulty of building complete repositories, or broad coverage grammars, etc).
2. The inconsistencies (either coming from the models, the knowledge or the speaker).
3. The complex interaction between different NLP levels (e.g. syntax and semantics [Yangarber+Grishman'98,Appelt+'96]).

Introduction

There are several inherent difficulties in most NLU tasks:

1. The open and compositional nature of language (i.e. the difficulty of building complete repositories, or broad coverage grammars, etc).
2. The inconsistencies (either coming from the models, the knowledge or the speaker).
3. The complex interaction between different NLP levels (e.g. syntax and semantics [Yangarber+Grishman'98,Appelt+'96]).
4. The combinatorial explosion of possibilities produced by all these issues.

13

Goals

Build a framework to:

- Explore the integration of knowledge

14

Goals

Build a framework to:

- Explore the integration of knowledge
- Explore the feasibility of integrating different NLP tasks

Goals

Build a framework to:

- Explore the integration of knowledge
- Explore the feasibility of integrating different NLP tasks
- Robustness through Optimization Techniques

Outline

NLP Architectures

- Reference Architectures: GATE [1], RAGS [Cahill+'99a], UIMA [2]

[1] http://gate.ac.uk
[2] http://www.ibm.com

NLP Architectures

- Reference Architectures: GATE [1], RAGS [Cahill+'99a], UIMA [2]
- Architectures can be classified according to:

[1] http://gate.ac.uk
[2] http://www.ibm.com

NLP Architectures

- Reference Architectures: GATE [1], RAGS [Cahill+'99a], UIMA [2]
- Architectures can be classified according to:
  - Integration of processes

[1] http://gate.ac.uk
[2] http://www.ibm.com

NLP Architectures

- Reference Architectures: GATE [1], RAGS [Cahill+'99a], UIMA [2]
- Architectures can be classified according to:
  - Integration of processes
  - Integration of knowledge

[1] http://gate.ac.uk
[2] http://www.ibm.com

NLP Architectures

- Reference Architectures: GATE [1], RAGS [Cahill+'99a], UIMA [2]
- Architectures can be classified according to:
  - Integration of processes
  - Integration of knowledge
- Integration versus modularity

[1] http://gate.ac.uk
[2] http://www.ibm.com

NLP Subtasks

/The/ /cat/ /eats/ /fish/

- **Tokenization**

NLP Subtasks

/My /sand cat/ /eats/ /fish/

- **Tokenization**
  - **Multiword Expressions**

NLP Subtasks

/Garfield the first/ /eats/ /fish/

- **Tokenization**
  - **Multiword Expressions**
  - **Named Entities**

25

NLP Subtasks

The$_{DT}$ *cat*$_{N}$ *eats*$_{V}$ *fish*$_{N}$

- **Tokenization**
  - **Multiword Expressions**
  - **Named Entities**
- **Part of Speech Tagging**

26

NLP Subtasks

the cat eat fish

- **Tokenization**
  - **Multiword Expressions**
  - **Named Entities**
- **Part of Speech Tagging**
- **Lemmatization**

NLP Subtasks

the cat#n#1 eats#v#2 fish#n#2

- **Tokenization**
  - **Multiword Expressions**
  - **Named Entities**
- **Part of Speech Tagging**
- **Lemmatization**
- **Word Sense Disambiguation (WSD)**

NLP Subtasks

$(the\ cat)_{NP}$ $(eats)_{VP}$ $(fish)_{NP}$

- **Tokenization**
  - **Multiword Expressions**
  - **Named Entities**
- **Part of Speech Tagging**
- **Lemmatization**
- **Word Sense Disambiguation (WSD)**
- **Parsing**

NLP Subtasks

(The cat)$_{agent}$ eats$_{event}$ fish$_{patient}$

- **Tokenization**
  - **Multiword Expressions**
  - **Named Entities**
- **Part of Speech Tagging**
- **Lemmatization**
- **Word Sense Disambiguation (WSD)**
- **Parsing**
- **Semantic Role Labelling** (SRL)

NLP Subtasks

I have a cat. It$_{cat}$ eats fish

- **Tokenization**
  - **Multiword Expressions**
  - **Named Entities**
- **Part of Speech Tagging**
- **Lemmatization**
- **Word Sense Disambiguation (WSD)**
- **Parsing**
- **Semantic Role Labelling** (SRL)
- **Anaphora Resolution**

Processes Integration

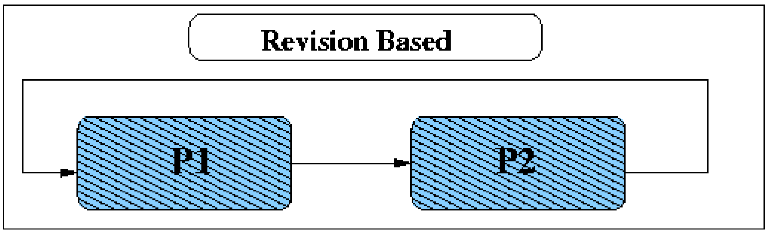


Figure: Revision Based Architecture

Processes Integration

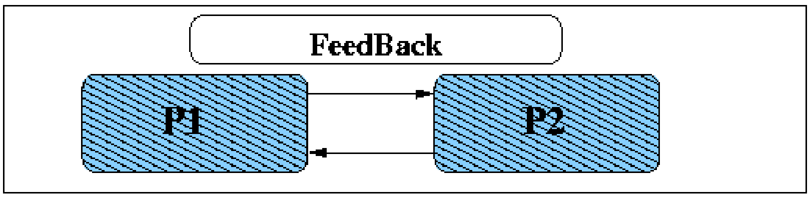


Figure: Feedback Architecture

Processes Integration

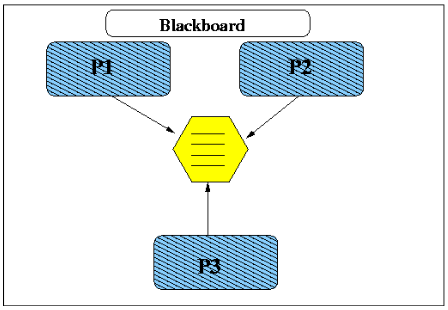


Figure: Blackboard Architecture

Outline

Knowledge Integration

- Automatically acquiring integrated information $\Longrightarrow$ **is difficult.**

Knowledge Integration

- Automatically acquiring integrated information $\Longrightarrow$ **is difficult.**
- There are many resources available:

Knowledge Integration

- Automatically acquiring integrated information $\Longrightarrow$ **is difficult.**
- There are many resources available:
  - sense repositories (WordNet, dictionaries),

Knowledge Integration

- Automatically acquiring integrated information $\Longrightarrow$ **is difficult.**
- There are many resources available:
  - sense repositories (WordNet, dictionaries),
  - ontologies (EuroWordNet Top Concept Ontology, SUMO),

39

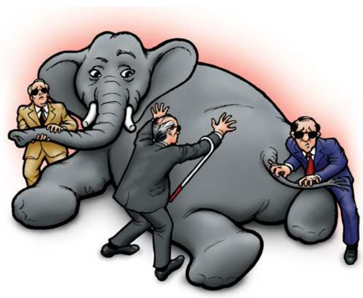
Knowledge Integration

- Automatically acquiring integrated information $\Longrightarrow$ **is difficult.**
- There are many resources available:
  - sense repositories (WordNet, dictionaries),
  - ontologies (EuroWordNet Top Concept Ontology, SUMO),
  - verbal subcategorization and selectional preferences information (VerbNet, PropBank, FrameNet)

40

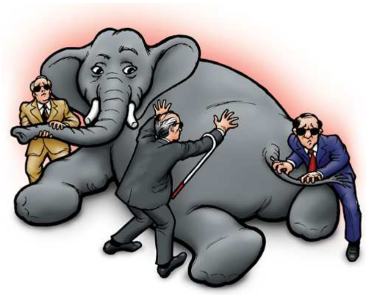
Knowledge Integration

- Each resource seems to capture some pieces of knowledge that the others do not



Knowledge Integration

- Each resource seems to capture some pieces of knowledge that the others do not
- and which could be crucial to solve a particular NLP task.



NLP Knowledge

Even if the integration is possible,

NLP Knowledge

Even if the integration is possible,
Can it be guaranted that the resulting resource will be either consistent, coherent or complete?

Outline

1. Introduction
   - Motivation
   - Process Integration for NLU
   - Knowledge Integration for NLU
2. **Knowledge Integration in PARDON**
3. PARDON's Architecture
   - Knowledge Representation in PARDON
   - Model Application
   - Inference Engine
   - Model Combination Constraints
4. Experiments
   - PARDON Semantic Role Labeler
   - PARDON Word Sense Disambiguator
5. Conclusions and Future Work
   - Conclusions
   - Further Work

Knowledge Integration in PARDON (MCR)

- Integrating already existing resources (mainly around the notion of 'sense')

Knowledge Integration in PARDON (MCR)

- Integrating already existing resources (mainly around the notion of 'sense')
- Keeping resource modularity

Knowledge Integration in PARDON (MCR)

- Integrating already existing resources (mainly around the notion of 'sense')
- Keeping resource modularity
- relating the different resources (checking consistence)

Knowledge Integration in PARDON (MCR)

- Integrating already existing resources (mainly around the notion of 'sense')
- Keeping resource modularity
- relating the different resources (checking consistence)
- but the real integration is performed when applying these resources for a particular piece of text

The Multilingual Central Repository

MCR follows the model proposed by the EuroWordNet project[3].

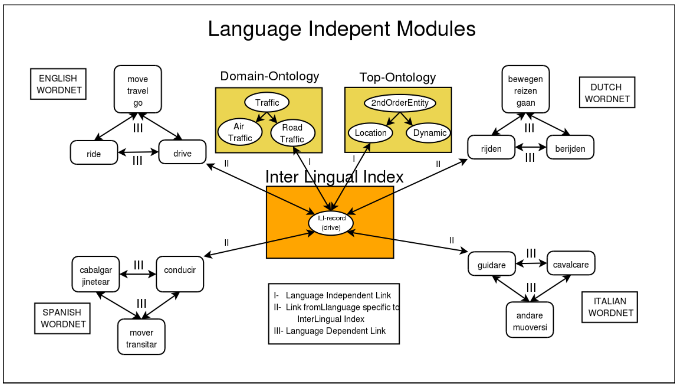


Figure: EuroWordNet architecture

[3] `http://www.illc.uva.nl/EuroWordNet`

The current MCR

- ILI aligned to WordNet 1.6
  - Base Concepts (EuroWordNet and BalkaNet)
  - EuroWordNet Top Concept Ontology
  - MultiWordNet Domains
  - Suggested Upper Merged Ontology
- Local wordnets:
  - Different versions of Princeton English WordNet
  - eXtended WordNet (XWN)
  - Basque, Catalan, Italian and Spanish wordnets
- Verb Lexicon (VerbNet)
- ...

The “Vaso” Example

vaso_1 02755829-n drinking_glass glass
GLOSS: a glass container for holding liquids while drinking
LF: 06-NOUN.ARTIFACT
DOMAIN: FACTOTUM
SUMO: Artifact+

TO: 1stOrderEntity-Form-Object
TO: 1stOrderEntity-Origin-Artifact
TO: 1stOrderEntity-Function-Container
TO: 1stOrderEntity-Function-Instrument

eXtended WordNet:
GLOSS: a glass#NN#2 container#NN#1 for hold#VBG#8 liquid#NNS#1 while drink#VBG#1

DOBJ SemCor

02755829 00849393-v 0.0074 polish shine smooth smoothen
02755829 00201878-v 0.0013 beautify embellish prettify
02755829 00826635-v 0.0010 get_hold_of take

WN2.0

RELATED TO: glass#v#4 (put in a glass container)

Figure: “Vaso” senses

The “Vaso” Example

vaso_2 04195626-n 04195626-n vessel vas:
GLOSS: a tube in which a body fluid circulates

LF: 08-NOUN.BODY
DOMAIN: ANATOMY
SUMO: BodyVessel+

TO: 1stOrderEntity-Form-Substance-Solid
TO: 1stOrderEntity-Origin-Natural-Living
TO: 1stOrderEntity-Composition-Part
TO: 1stOrderEntity-Function-Container

eXtended WordNet:
GLOSS: a tube#NN#4 in which a body_fluid#NN#1 circulate#VBZ#4

DOBJ SemCor

04195626 01781222 0.0334 be occur
04195626 00058757 0.0072 inject shoot
04195626 01357963 0.0068 follow travel_along
04195626 00055849 0.0045 administer dispense
04195626 01012352 0.0022 block close_up impede jam obstruct occlude

SUBJ SemCor

04195626 01831830 0.0133 stop terminate

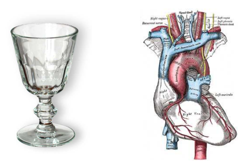


Figure: “Vaso” senses

The "Vaso" Example

vaso_3 09914390-n glassful glass:
GLOSS: the quantity a glass will hold

LF: 23-NOUN.QUANTITY
DOMAIN: NUMBER
SUMO: ConstantQuantity+

TO: 1stOrderEntity-Composition-Part
TO: 2ndOrderEntity-SituationType-Static
TO: 2ndOrderEntity-SituationComponent-Quantity

eXtended WordNet:
GLOSS: the quantity#NN#1 a glass#NN#2 will hold#VB#1

DOBJ SemCor

09914390 00795711 0.0026 drink imbibe
09914390 01530096 0.0009 accept have take
09914390 00786286 0.0009 consume have ingest take
take_in 09914390 01513874 0.0001 acquire get

DOBJ Semcor No generalization

09914390 00795711 drink imbibe

09914390 01530096 accept have take

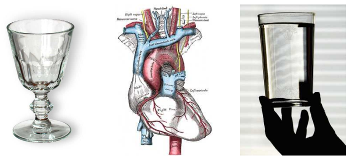


Figure: "Vaso" senses

Outline

1. Introduction
   - Motivation
   - Process Integration for NLU
   - Knowledge Integration for NLU
2. Knowledge Integration in PARDON
3. PARDON's Architecture
   - Knowledge Representation in PARDON
   - Model Application
   - Inference Engine
   - Model Combination Constraints
4. Experiments
   - PARDON Semantic Role Labeler
   - PARDON Word Sense Disambiguator
5. Conclusions and Future Work
   - Conclusions
   - Further Work

PARDON's Architecture

- General framework to easily formalize different NLP tasks.

PARDON's Architecture

- General framework to easily formalize different NLP tasks.
- These different tasks can be tested separately or carried out simultaneously.

PARDON's Architecture

- General framework to easily formalize different NLP tasks.
- These different tasks can be tested separately or carried out simultaneously.
- Without over-constraining the interaction between processes/knowledge.

PARDON's Architecture

- General framework to easily formalize different NLP tasks.
- These different tasks can be tested separately or carried out simultaneously.
- Without over-constraining the interaction between processes/knowledge.
- Using Consistent Labelling Problem (CLP) as the framework to:

PARDON's Architecture

- General framework to easily formalize different NLP tasks.
- These different tasks can be tested separately or carried out simultaneously.
- Without over-constraining the interaction between processes/knowledge.
- Using Consistent Labelling Problem (CLP) as the framework to:
  - integrate different NLP processes and

PARDON's Architecture

- General framework to easily formalize different NLP tasks.
- These different tasks can be tested separately or carried out simultaneously.
- Without over-constraining the interaction between processes/knowledge.
- Using Consistent Labelling Problem (CLP) as the framework to:
  - integrate different NLP processes and
  - to apply any kind of knowledge (syntactic, semantic, linguistic, statistical) at the earliest opportunity,

PARDON's Architecture

- General framework to easily formalize different NLP tasks.
- These different tasks can be tested separately or carried out simultaneously.
- Without over-constraining the interaction between processes/knowledge.
- Using Consistent Labelling Problem (CLP) as the framework to:
  - integrate different NLP processes and
  - to apply any kind of knowledge (syntactic, semantic, linguistic, statistical) at the earliest opportunity,
  - while retaining an independent representation for every kind of knowledge.

Consistent Labelling Problem (CLP)

- A Labelling Problem is defined by a set of variables V_i, a set of labels (domain) for each variable D_i, a compatibility relation over tuples.

$v_{1,1}$	$v_{1,2}$	$v_{1,3}$	$\ldots$					
$v_{2,1}$								
$v_{3,1}$								
$\ldots$								

Figure: Sudoku game

Consistent Labelling Problem (CLP)

- Compatibilities are real-value functions $r_{ij} : DxD \longrightarrow \Re$ where $r_{i,j}(a, b)$ refers to the compatibility of the simultaneous assignment of a to V_i and b to V_j.

Var.	**Values**
$v_{1,1}$	{ 1 }
$v_{1,2}$	{ 1,2,3,4,5, 6,7,8,9 }
$v_{1,3}$	{ 1,2,3,4,5, 6,7,8,9 }
...	
$v_{10,10}$	{ 1,2,3,4,5, 6,7,8,9 }

1								

Consistent Labelling Problem (CLP)

- Compatibilities are real-value functions $r_{ij} : DxD \longrightarrow \Re$ where $r_{i,j}(a, b)$ refers to the compatibility of the simultaneous assignment of a to V_i and b to V_j.

$$v_{1,1} = 1 \nsim v_{1,2} = 1$$

Var.	**Values**
$v_{1,1}$	{ 1 }
$v_{1,2}$	{ 1,2,3,4,5, 6,7,8,9 }
$v_{1,3}$	{ 1,2,3,4,5, 6,7,8,9 }
...	
$v_{10,10}$	{ 1,2,3,4,5, 6,7,8,9 }

1	1							

Consistent Labelling Problem (CLP)

- Compatibilities are real-value functions $r_{ij} : DxD \longrightarrow \Re$ where $r_{i,j}(a, b)$ refers to the compatibility of the simultaneous assignment of a to V_i and b to V_j.

$$v_{1,1} = 1 \not\sim v_{1,3} = 1$$

Var.	Values
$v_{1,1}$	{ 1 }
$v_{1,2}$	{ 1,2,3,4,5, 6,7,8,9 }
$v_{1,3}$	{ 1,2,3,4,5, 6,7,8,9 }
...	
$v_{10,10}$	{ 1,2,3,4,5, 6,7,8,9 }

1		1						

Consistent Labelling Problem (CLP)

- Compatibilities are real-value functions $r_{ij} : DxD \longrightarrow \Re$ where $r_{i,j}(a, b)$ refers to the compatibility of the simultaneous assignment of a to V_i and b to V_j.

$$v_{1,1} = 1 \not\sim v_{1,4} = 1$$

Var.	Values
$v_{1,1}$	{ 1 }
$v_{1,2}$	{ 1,2,3,4,5, 6,7,8,9 }
$v_{1,3}$	{ 1,2,3,4,5, 6,7,8,9 }
...	
$v_{10,10}$	{ 1,2,3,4,5, 6,7,8,9 }

1			1					

Consistent Labelling Problem (CLP)

- Compatibilities are real-value functions $r_{ij} : DxD \longrightarrow \Re$ where $r_{i,j}(a,b)$ refers to the compatibility of the simultaneous assignment of a to V_i and b to V_j.

$$v_{1,1} = 1 \nsim v_{1,5} = 1$$

Var.	Values
$v_{1,1}$	{ 1 }
$v_{1,2}$	{ 1,2,3,4,5, 6,7,8,9 }
$v_{1,3}$	{ 1,2,3,4,5, 6,7,8,9 }
...	
$v_{10,10}$	{ 1,2,3,4,5, 6,7,8,9 }

1				1				

Consistent Labelling Problem (CLP)

- Compatibilities are real-value functions $r_{ij} : DxD \longrightarrow \Re$ where $r_{i,j}(a,b)$ refers to the compatibility of the simultaneous assignment of a to V_i and b to V_j.

$$v_{1,1} = 1 \nsim v_{1,6} = 1$$

Var.	Values
$v_{1,1}$	{ 1 }
$v_{1,2}$	{ 1,2,3,4,5, 6,7,8,9 }
$v_{1,3}$	{ 1,2,3,4,5, 6,7,8,9 }
...	
$v_{10,10}$	{ 1,2,3,4,5, 6,7,8,9 }

1					1			

Consistent Labelling Problem (CLP)

- Compatibilities are real-value functions $r_{ij} : DxD \longrightarrow \Re$ where $r_{i,j}(a, b)$ refers to the compatibility of the simultaneous assignment of a to V_i and b to V_j.

$$v_{1,1} = 1 \nsim v_{1,7} = 1$$

Var.	**Values**
$v_{1,1}$	{ 1 }
$v_{1,2}$	{ 1,2,3,4,5, 6,7,8,9 }
$v_{1,3}$	{ 1,2,3,4,5, 6,7,8,9 }
...	
$v_{10,10}$	{ 1,2,3,4,5, 6,7,8,9 }

1						1		

Consistent Labelling Problem (CLP)

- Compatibilities are real-value functions $r_{ij} : DxD \longrightarrow \Re$ where $r_{i,j}(a, b)$ refers to the compatibility of the simultaneous assignment of a to V_i and b to V_j.

$$v_{1,1} = 1 \nsim v_{1,8} = 1$$

Var.	**Values**
$v_{1,1}$	{ 1 }
$v_{1,2}$	{ 1,2,3,4,5, 6,7,8,9 }
$v_{1,3}$	{ 1,2,3,4,5, 6,7,8,9 }
...	
$v_{10,10}$	{ 1,2,3,4,5, 6,7,8,9 }

1							1	

Consistent Labelling Problem (CLP)

- Compatibilities are real-value functions $r_{ij} : DxD \longrightarrow \Re$ where $r_{i,j}(a,b)$ refers to the compatibility of the simultaneous assignment of a to V_i and b to V_j.

$$v_{1,1} = 1 \not\sim v_{1,9} = 1$$

Var.	Values
$v_{1,1}$	{ 1 }
$v_{1,2}$	{ 1,2,3,4,5, 6,7,8,9 }
$v_{1,3}$	{ 1,2,3,4,5, 6,7,8,9 }
...	
$v_{10,10}$	{ 1,2,3,4,5, 6,7,8,9 }

1								1

Relaxation Labelling

- Consistent Labelling Problems (CLP) can be solved via Relaxation Labelling
- Relaxation labelling is a generic name for a family of iterative algorithms which perform function optimization.
- The algorithm finds a combination of values for a set of variables such that satisfies -to a maximum possible degree- a set of given constraints.
- This formulation allows to naturally integrate different kinds of knowledge coming from different sources (linguistic and statistical), which may be partial, partially incorrect or even inconsistent.

PARDON's Architecture

Thus, PARDON's Architecture is similar to a rule-based system and has three main components:

- **Knowledge Representation**: How the information, either for partial analysis or for the whole sentence, is represented.

PARDON's Architecture

Thus, PARDON's Architecture is similar to a rule-based system and has three main components:

- **Knowledge Representation**: How the information, either for partial analysis or for the whole sentence, is represented.
- **Model Application**: Under which conditions and how a model is applied. How the several parts of the model/pattern could be identified in the input.

PARDON's Architecture

Thus, PARDON's Architecture is similar to a rule-based system and has three main components:

- **Knowledge Representation**: How the information, either for partial analysis or for the whole sentence, is represented.
- **Model Application**: Under which conditions and how a model is applied. How the several parts of the model/pattern could be identified in the input.
- **Inference Engine**: How and when it is decided to apply a model.

Outline

Knowledge Representation in PARDON

$$\begin{bmatrix} \textit{c2} & & \\ \text{HEAD} & \text{EAT} & \\ \text{POS} & \text{VVB} & \\ \text{CATG} & \text{VP} & \\ \text{SENSE} & \text{EAT\#N\#1} & \\ \text{MODEL} & \text{TRANSITIVE} & \\ \text{AGENT} & \begin{bmatrix} \textit{c1} & \\ \text{HEAD} & \text{CAT} \\ \text{SENSE} & \text{CAT\#N\#1} \end{bmatrix} & \\ \text{PATIENT} & \begin{bmatrix} \textit{c3} & \\ \text{HEAD} & \text{FISH} \\ \text{SENSE} & \text{FISH\#N\#2} \end{bmatrix} & \end{bmatrix}$$

Variable	**Values**
C1.HEAD	{ cat }
C1.SENSE	{ cat#n#1 }
C2.HEAD	{ fish }
C2.SENSE	{ fish#n#2 }
C3.POS	{ VVB }
C3.CATG	{ VP }
C3.HEAD	{ eat }
C3.SENSE	{ eat#v#1 }
C3.MODEL	{ transitive }
C3.AGENT	{ c1 }
C3.PATIENT	{ c2 }

Figure: Variables associated with the frame-like representation of *cat*

Outline

Role and Model Application

Lexicon	
Word	**PoS**
cat	N, V
eat	V
fish	N, V

CFG Grammar		
Head	**Id**	**CFG Rule**
N	MNP	D, N $\Longrightarrow$ NP
V	MS	NP_1, V, NP_2 $\Longrightarrow$ S

Figure: A simple Context Free Grammar

Model Application Constraints

- **Compulsory**: The object attribute must match the role attribute.

Model Application Constraints

- **Compulsory**: The object attribute must match the role attribute.
- **Optional**: The object will be considered as a possible filler of the role, even if, the object attribute does not match the role attribute.

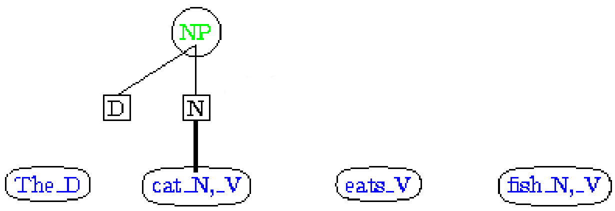
Model Application Constraints

- **Compulsory**: The object attribute must match the role attribute.
- **Optional**: The object will be considered as a possible filler of the role, even if, the object attribute does not match the role attribute.
- **Ignore**: The object could contain information that the matching function should not take into account.

Model Application Constraints

Variable	Values	Variable	Values
C1.ROLE	{ }	C3.ROLE	{ }
C1.MODEL	{ }	C3.MODEL	{ }
C2.ROLE	{ }	C4.ROLE	{ }
C2.MODEL	{ }	C4.MODEL	{ }

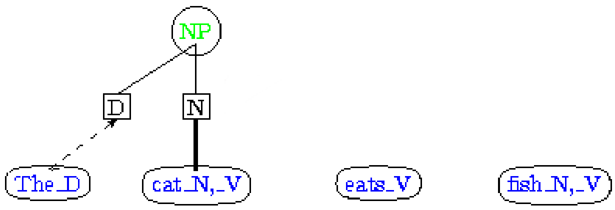
Model Application Constraints



Variable	Values	Variable	Values
C1.ROLE	{ }	C3.ROLE	{ }
C1.MODEL	{ }	C3.MODEL	{ }
C2.ROLE	{ }	C4.ROLE	{ }
C2.MODEL	{ MNP }	C4.MODEL	{ }

85

Model Application Constraints

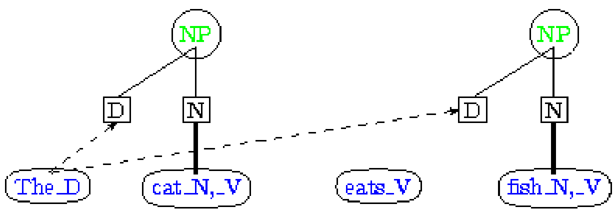


Variable	Values
C1.ROLE	{D.MNP.c2 }
C1.MODEL	{ }
C2.ROLE	{ }
C2.MODEL	{ MNP }

Variable	Values
C3.ROLE	{ }
C3.MODEL	{ }
C4.ROLE	{ }
C4.MODEL	{ }

86

Model Application Constraints

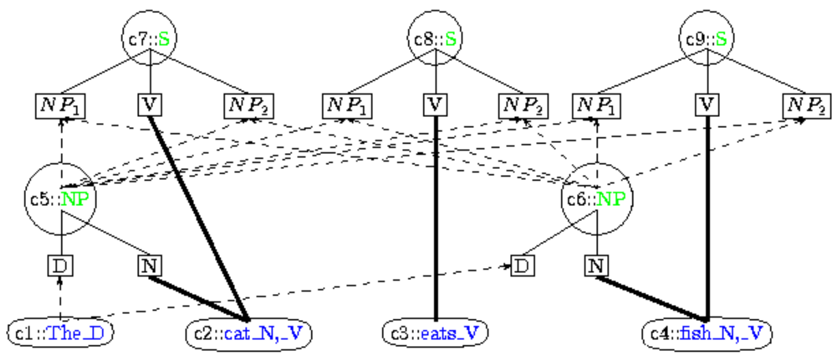


Variable	Values
C1.ROLE	{D.MNP.c2 ,D.MNP.c4}
C1.MODEL	{ }
C2.ROLE	{ }
C2.MODEL	{ MNP }

Variable	Values
C3.ROLE	{ }
C3.MODEL	{ }
C4.ROLE	{ MNP }
C4.MODEL	{ }

Outline

Derivational Sequences



Outline

Object Uniqueness

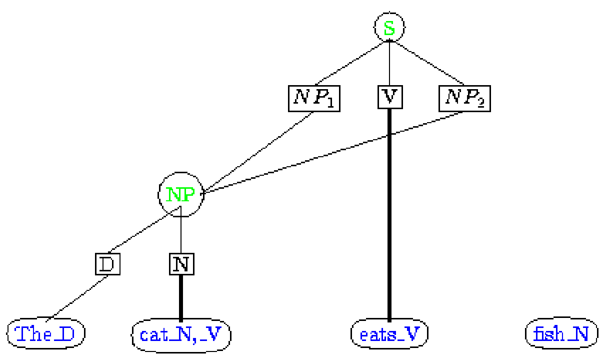


Figure: "The cat eats the cat"+"fish"

Role Uniqueness

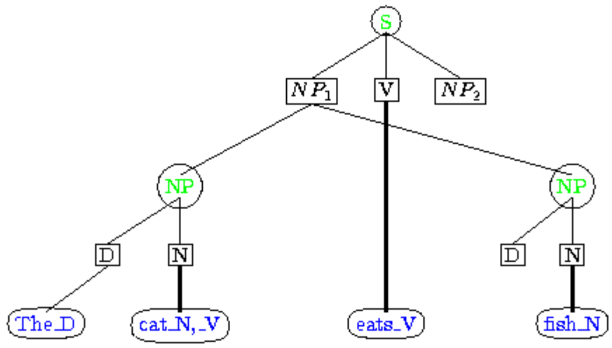


Figure: "The cat+fish eats something"

Model Uniqueness

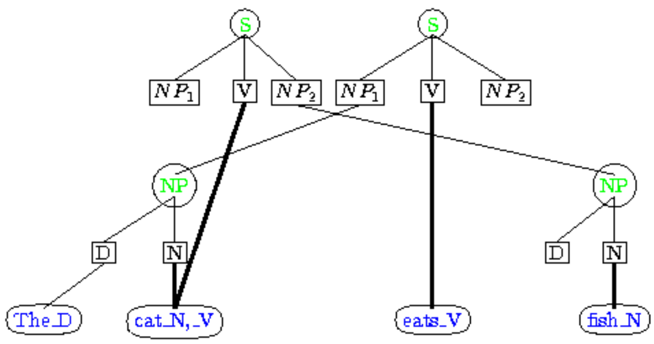


Figure: "The cat eats something"+"somebody cats fish"

Role Inconsitence

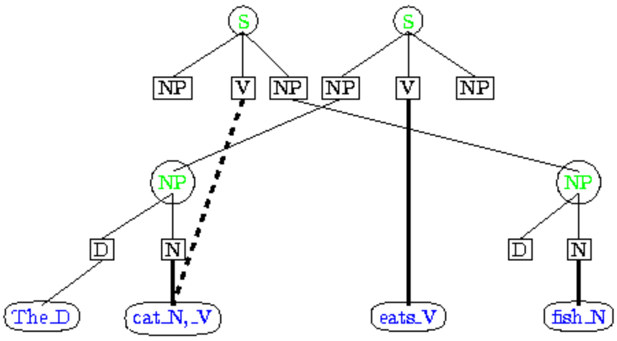


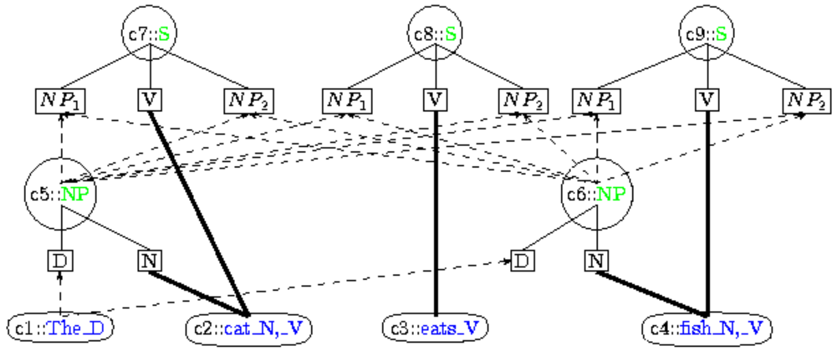
Figure: "The cat eats something"+"something cats fish"

Structural Constraints

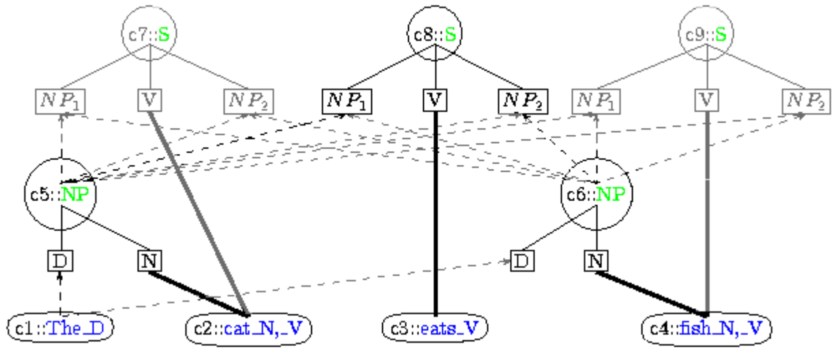
A set of axioms is needed to ensure the correctness of the partial object combination (**structural constraints**).

- **TOP Uniqueness**
- **TOP Existence**
- **No Cycles**
- **NONE Support**

Iterative Inference Engine

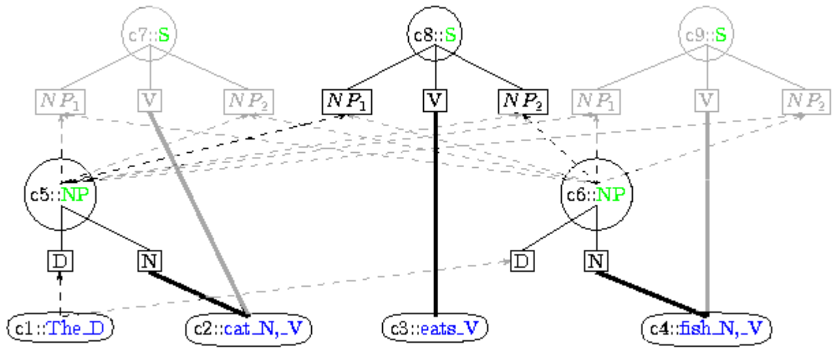


Iterative Inference Engine



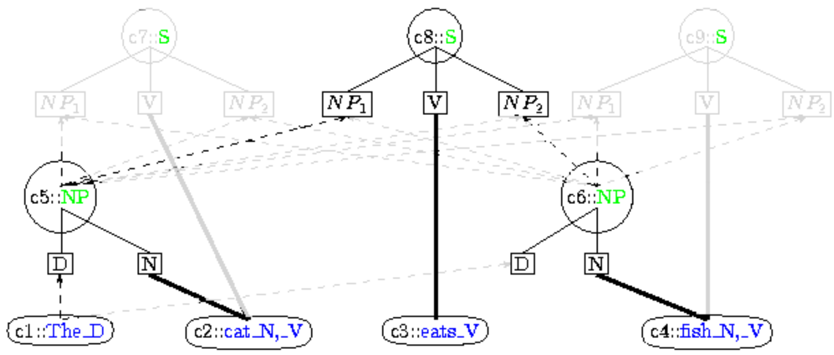
97

Iterative Inference Engine

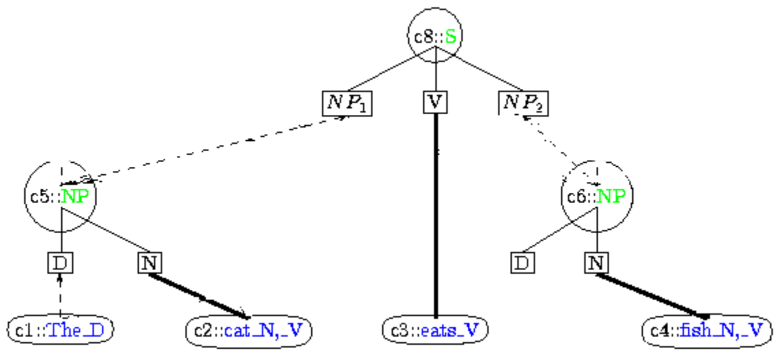


98

Iterative Inference Engine



Iterative Inference Engine



Consistent Partial Objects

Object	**Needs**
D:(NONE) N:(cat)	opt.
D:(The) N:(cat)	-
...	
NP_1: NONE V:eat NP_2: NONE	opt.
NP_1: (D:(NONE) N:(cat)) V:eat NP_2: NONE	opt.
NP_1: (D:(The) N:(cat)) V:eat NP_2: NONE	
NP_1: NONE V:eat NP_1: (D:(NONE) N:(fish))	
NP_1: NONE V:eat NP_1: (D:(The) N:(fish))	
NP_1: (D:(NONE) N:(cat)) V:eat NP_2: (D:(NONE) N:(fish))	opt.
NP_1: (D:(NONE) N:(cat)) V:eat NP_2: (D:(The) N:(fish))	opt+gap+un
...	

Figure: Consistent Partial Objects generated from "The cat eats fish"

Amalgamating the Search Space

- 196 different parse trees for our example sentence,
- Reducing the search space is an important issue!!
- In order to soften this combinatorial explosion, given an initial object,
- we will amalgamate the representation of all the possible objects which could be generated using the models associated to the same initial object.
- Thus, while an object uses its models to combine itself with other objects, some of the resulting object values are determined (in a similar way to Hirst's *Polaroid Words* [Hirst'87]).

Iterative WSD

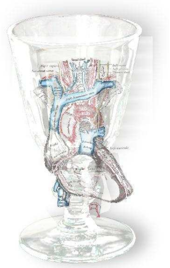


Figure: Different states during WSD

103

Iterative WSD

Figure: Different states during WSD

104

Iterative WSD

Figure: Different states during WSD

Iterative WSD

Figure: Different states during WSD

The Amalgamated Representation

Variable	Values
C1.CATG	{ D }
C1.ROLE	{ TOP, D.MNP.c2, D.MNP.c4 }
C1.MOD	{ NONE }
C2.CATG	{ N, V, NP, S }
C2.ROLE	{ TOP, NP_1.MS.c2, NP_2.MS.c2, NP_1.MS.c3, NP_2.MS.c3 }
C2.MOD	{ NONE, MNP, MS }
C3.CATG	{ V, S }
C3.ROLE	{ TOP }
C3.MOD	{ NONE, MNP, MS }
C4.CATG	{ N, V, NP, S }
C4.ROLE	{ TOP, NP_1.MS.c2, NP_2.MS.c2, NP_1.MS.c3, NP_2.MS.c3 }
C4.MOD	{ NONE, MNP, MS }

Attribute Propagation/Percolation

More complex models, such as the models which include a re-entrancy, need a more complex representation.

"$(The\ group)_{c1}$ $(of\ hooligans)_{c2}$ ..."

$$[c1.model = group \wedge \quad c2.role = (member, group, c1)\ ] \quad \sim \quad c1.sem = c2.sem$$

Expanding into:

$$c1.model = group \wedge \quad c2.role = (member, group, c1) \wedge \quad c2.sem = Human \quad \sim \quad c1.sem = Human$$

Outline

Outline

Semantic Parsing

El gato	*come*	*pescado*
The cat	eats	fish
Starter		**Entity**

Figure: Example of semantic roles.

LEXPIR

- PIRAPIDES [Vazquez+'00] was a project that focussed on the study of the English, Spanish and Catalan verbal predicates.
- Verb classes are organized in a hierarchy which enables the use of default monotonic inheritance to describe verb properties.
- Capturing the argumental structure of a Spanish sentence may be a hard task:
  - given the optionality of some constituents and
  - the free word–order syntax structure of Spanish.

LEXPIR Models

basic model for Trajectory verbs					
Catg.	Handle	**Comp.**	Sem.	Agree.	Opt.
NP	p_inic	**starter**	Human	yes	yes
x	x	**entity**	Top	no	yes
PP	p_ruta	**route**	Top	no	yes
PP	p_orig	**source**	Top	no	yes
PP	x	**destination**	Top	no	yes

Table: Basic Model for trajectory verbs

113

LEXPIR Models

basic model for Trajectory verbs					
Catg.	Handle	Comp.	Sem.	Agree.	Opt.
NP	p_inic	starter	Human	yes	yes
x	x	entity	Top	no	yes
PP	p_ruta	route	Top	no	yes
PP	p_orig	source	Top	no	yes
PP	x	destination	Top	no	yes

Table: Basic Model for trajectory verbs

114

LEXPIR Models

basic model for Trajectory verbs					
Catg.	**Handle**	Comp.	Sem.	Agree.	Opt.
NP	**p_inic**	starter	Human	yes	yes
x	**x**	entity	Top	no	yes
PP	**p_ruta**	route	Top	no	yes
PP	**p_orig**	source	Top	no	yes
PP	**x**	destination	Top	no	yes

Table: Basic Model for trajectory verbs

LEXPIR Models

basic model for Trajectory verbs					
Catg.	Handle	Comp.	**Sem.**	Agree.	Opt.
NP	p_inic	starter	**Human**	yes	yes
x	x	entity	**Top**	no	yes
PP	p_ruta	route	**Top**	no	yes
PP	p_orig	source	**Top**	no	yes
PP	x	destination	**Top**	no	yes

Table: Basic Model for trajectory verbs

LEXPIR Models

basic model for Trajectory verbs					
Catg.	Handle	Comp.	Sem.	**Agree.**	Opt.
NP	p_inic	starter	Human	**yes**	yes
x	x	entity	Top	**no**	yes
PP	p_ruta	route	Top	**no**	yes
PP	p_orig	source	Top	**no**	yes
PP	x	destination	Top	**no**	yes

Table: Basic Model for trajectory verbs

LEXPIR Models

basic model for Trajectory verbs					
Catg.	Handle	Comp.	Sem.	Agree.	**Opt.**
NP	p_inic	starter	Human	yes	**yes**
x	x	entity	Top	no	**yes**
PP	p_ruta	route	Top	no	**yes**
PP	p_orig	source	Top	no	**yes**
PP	x	destination	Top	no	**yes**

Table: Basic Model for trajectory verbs

Formalization

(este año)$_{NP}$ (en el congreso)$_{PP}$ (del partido)$_{PP}$ (se)$_{PR}$ (habló)$_{VP}$ (de las pensiones)$_{PP}$

	Variable Name	Values
este año	*c1*.model	N$_{de}$ NONE
	c1.role	*(starter, basic, c5)* TOP
en el congreso	*c2*.model	N$_{de}$ NONE
	c2.role	TOP
del partido	*c3*.model	N$_{de}$ NONE
	c3.role	*(entity, basic, c5)* *(entity, impersonal, c5)* *(modif, N$_{de}$, c1)* *(modif, N$_{de}$, c2)* *(modif, N$_{de}$, c3)* TOP

	Variable Name	Values
se	*c4*.model	NONE
	c4.role	*(se, impersonal, c5)* TOP
habló	*c5*.model	basic impersonal NONE
	c5.role	TOP
de las pensiones	*c6*.model	N$_{de}$ NONE
	c6.role	*(entity, basic, c5)* *(entity, impersonal, c5)* *(modif, N$_{de}$, c1)* *(modif, N$_{de}$, c2)* *(modif, N$_{de}$, c3)* TOP

Figure: CLP associated

Experiments

- 170 real sentences were taken from a Spanish newspaper
- labelled by hand with their verbal models and meaning components.
- The sentence average length is 8.1 words, ranging from 3 to 23.
- Only one-verb sentences were selected.

Results

COR	*INC*	*PRE*	*REC*
155	8	**95%**	**91%**

Table: Verbal Model identification results

121

Results

COR	*INC*	*PRE*	*REC*
155	8	95%	91%

Table: Verbal Model identification results

COR	*INC*	*MIS*	*SPU*	*POS*	*ACT*
203	27	60	51	290	281

UND	*OVR*	*SUB*	*ERR*	*PRE*	*REC*
20%	18%	12%	40%	72%	70%

P&R	*2P&R*	*P&2R*
71%	70%	72%

Table: Verbal case-role filling results

122

Results

COR	*INC*	*PRE*	*REC*
155	8	95%	91%

Table: Verbal Model identification results

COR	*INC*	*MIS*	*SPU*	*POS*	*ACT*
203	27	60	51	290	281

UND	*OVR*	*SUB*	*ERR*	*PRE*	*REC*
20%	18%	12%	40%	**72%**	**70%**

P&R	*2P&R*	*P&2R*
71%	70%	72%

Table: Verbal case-role filling results

Results

COR	INC	PRE	REC
155	8	95%	91%

Table: Verbal Model identification results

COR	INC	MIS	SPU	POS	ACT
203	27	60	51	290	281

UND	OVR	SUB	ERR	PRE	REC
20%	**18**%	12%	40%	72%	70%

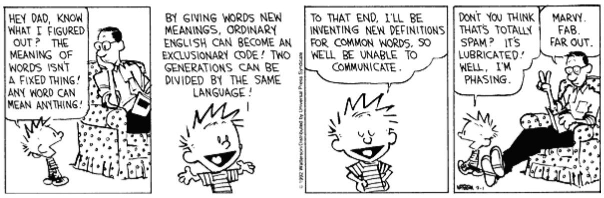
P&R	2P&R	P&2R
71%	70%	72%

Table: Verbal case-role filling results

Outline

WSD

- WSD can be defined as the process of deciding the meaning of a word in its context. The possible senses for a word are previously defined in a sense repository (usually WordNet).



WSD

- Each word sense would have associated a set of models with syntactic and semantic information.

WSD

- Each word sense would have associated a set of models with syntactic and semantic information.
- Our task will consist in establishing which of these models is more similar to the input sentence.

WSD

- Each word sense would have associated a set of models with syntactic and semantic information.
- Our task will consist in establishing which of these models is more similar to the input sentence.
- The syntactic-semantic model selected will establish the correct word sense,

WSD

- Each word sense would have associated a set of models with syntactic and semantic information.
- Our task will consist in establishing which of these models is more similar to the input sentence.
- The syntactic-semantic model selected will establish the correct word sense,
- not only for the syntactic head but also, if possible, for the rest of the roles.

SENSEVAL-II English Lexical sample task

- The task consists in disambiguating the occurrences of 73 different words (noun, verbs and adjectives)
- Test corpus of 4,328 paragraphs.
- Verbal WordNet senses were not directly used in SENSEVAL-III.

SENSEVAL-II English Lexical Task

- Models from Frament-Verbnet, only cover 50 senses of the test (i.e. 640 out of 4,328)
- Learn the models automaticaly
- Example-based learning: LEXAS [Ng+Lee'1996], TIMBLE [HKD01], GAMBL [DHDV04].

Obtaining Lexical Models

As presented by Mr. Chabrol, and <**head**>*played*<**/head**> with thin-lipped intensity by Isabelle Huppert, Marie-Louise (called Marie Latour in the **film**) was not a nice person.

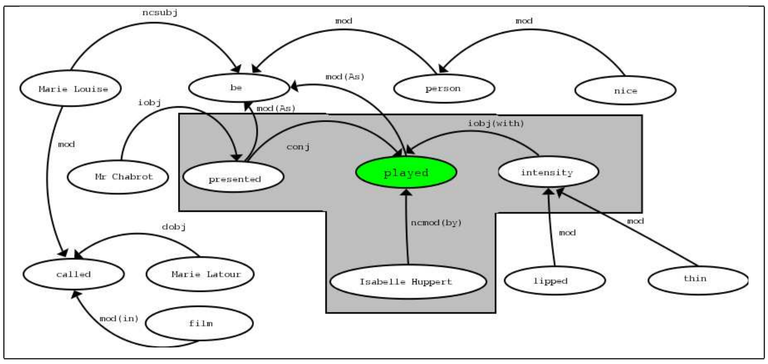


Figure: Dependency Analysis obtained using RASP

Obtaining Lexical Models

	#models	#models senses-in-test	#sense head	#sense role
SemCor	**246,083**	1,015	667	348
Senseval	**75,707**	13,068	6,073	6,995

Table: Models acquired for the 73 words included in SENSEVAL-II test corpus

Obtaining Lexical Models

	#models	#models senses-in-test	#sense head	#sense role
SemCor	246,083	**1,015**	667	348
Senseval	75,707	**13,068**	6,073	6,995

Table: Models acquired for the 73 words included in SENSEVAL-II test corpus

Head Supervised WSD

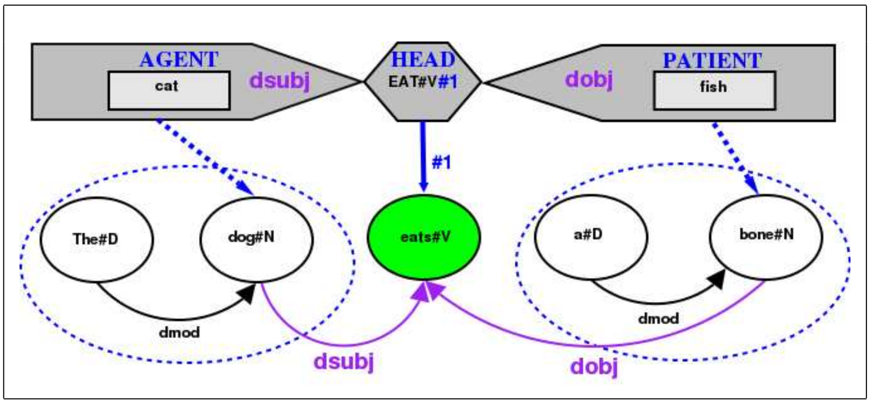


Figure: Model Matching

Role Supervised WSD

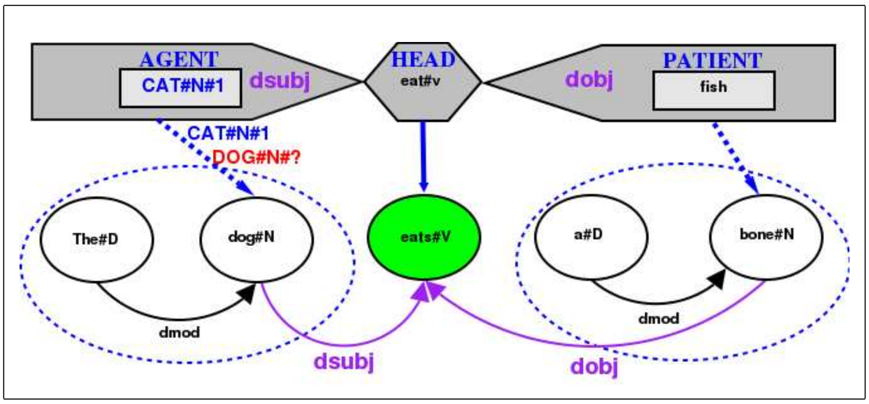


Figure: Role Matching

Role Unsupervised WSD

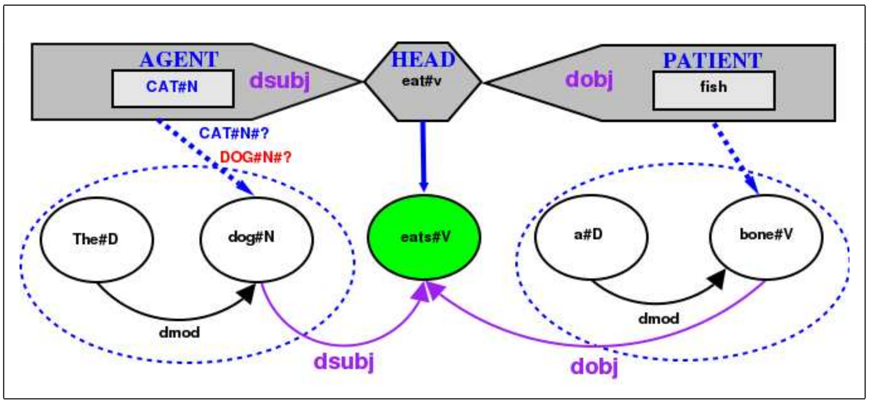


Figure: Role Matching

Head Unsupervised WSD

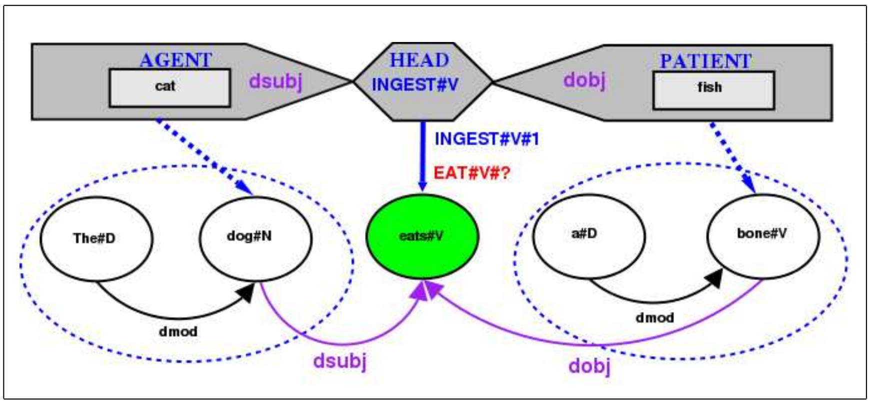


Figure: Model Matching

Senseval-II Evaluation Issues

There are several issues that can mislead our evaluation of the SENSEVAL-II English Lexical sample task [Escudero'06]:

- **MWEs**: The systems have to tokenize and recognise multiwords.

Senseval-II Evaluation Issues

There are several issues that can mislead our evaluation of the SENSEVAL-II English Lexical sample task [Escudero'06]:

- **MWEs**: The systems have to tokenize and recognise multiwords.
- **Special votes**: The systems were allowed to vote for ProperNouns (P) and Unknown senses (U).

Senseval-II Evaluation Issues

There are several issues that can mislead our evaluation of the SENSEVAL-II English Lexical sample task [Escudero'06]:

- **MWEs**: The systems have to tokenize and recognise multiwords.
- **Special votes**: The systems were allowed to vote for ProperNouns (P) and Unknown senses (U).
- **Inconsistencies in the training data**: There are a few inconsistencies in the data.

Senseval-II Evaluation Issues

There are several issues that can mislead our evaluation of the SENSEVAL-II English Lexical sample task [Escudero'06]:

- **MWEs**: The systems have to tokenize and recognise multiwords.
- **Special votes**: The systems were allowed to vote for ProperNouns (P) and Unknown senses (U).
- **Inconsistencies in the training data**: There are a few inconsistencies in the data.
- **Processing issues**: There are quite a few typographical tags in the training corpus which makes the preprocessing (specially the syntactic analysis of the text) much more difficult.

Baseline WSD

	Fine			Coarse		
	P	**R**	**F1**	**P**	**R**	**F1**
PARDON-MFS Training	47.0	46.2	46.6	53.9	53.1	53.5
PARDON-MFS SemCor	40.8	40.1	40.4	50.0	49.0	49.5

Table: Baseline using MFS for SENSEVAL-II English Lexical Sample task

Upper Bounds

Strategy	**Upper Bound**
head supervised	**49%** (2,122)
head (pre-process)	71% (3,081)
head-role supervised	60% (2,628)
head-role (pre-process)	87% (3,793)

Table: Upper Bounds using the SENSEVAL-II Training

Upper Bounds

Strategy	Upper Bound
head supervised	49% (2,122)
head (pre-process)	71% (3,081)
head-role supervised	**60%** (2,628)
head-role (pre-process)	87% (3,793)

Table: Upper Bounds using the SENSEVAL-II Training

Upper Bounds

Strategy	Upper Bound
head supervised	49% (2,122)
head (pre-process)	71% (3,081)
head-role supervised	60% (2,628)
head-role (pre-process)	**87%** (3,793)

Table: Upper Bounds using the SENSEVAL-II Training

The More I Know the Better I WSD?

Semantics					Fine			Coarse		
LF	**Wn**	**Sumo**	**Dom**	**Tco**	**P**	**R**	**F1**	**P**	**R**	**F1**
					37.0	26.2	30.7	47.1	33.4	39.1
×					37.0	26.8	31.1	47.2	34.2	39.7
	×				38.6	27.4	32.0	48.7	34.7	40.5
		×			38.7	27.4	32.1	48.6	34.5	40.3
			×		42.5	29.8	35.0	52.5	36.9	43.3
				×	42.8	30.1	35.3	53.0	37.2	44.0
×	×				43.8	31.8	36.8	53.3	38.7	44.8
×	×	×			43.6	31.7	36.7	53.3	38.7	44.8
×	×	×	×		43.5	31.6	36.5	53.3	38.7	44.8
×	×	×	×	×	41.8	29.3	34.4	51.6	36.1	42.5

Table: Results for SENSEVAL-II English Lexical Sample task

Comparing with the SENSEVAL-II English lexical sample task participants

	Verbs			Nouns			Adjectives		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
JHU(R)	56.6	56.6	56.6	68.2	68.2	68.2	73.2	73.2	73.2
SMUIs	56.3	56.3	56.3	69.5	6.95	69.5	66.8	66.8	66.8
KUNLP	57.6	57.6	57.6	66.8	66.8	66.8	66.8	66.8	66.8
CS224n	52.3	52.3	52.3	68.3	68.3	68.3	61.6	61.7	61.6
Sinequa	53.5	53.5	53.5	63.3	63.3	63.3	66.4	66.4	66.4
Talp	51.3	51.3	51.3	65.5	65.5	65.5	64.5	64.5	64.5
Pardon mfs	**46.4**	40.1	43.0	55.7	53.6	54.6	60.7	54.2	57.3
Pardon ufs	47.1	37.7	41.9	47.9	34.3	38.7	47.4	31.5	35.2

Table: Results in **P**recision and **R**ecall for each PoS

Comparing with the SENSEVAL-II English lexical sample task participants

	Verbs			Nouns			Adjectives		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
JHU(R)	56.6	56.6	56.6	68.2	68.2	68.2	73.2	73.2	73.2
SMUIs	56.3	56.3	56.3	69.5	6.95	69.5	66.8	66.8	66.8
KUNLP	57.6	57.6	57.6	66.8	66.8	66.8	66.8	66.8	66.8
CS224n	52.3	52.3	52.3	68.3	68.3	68.3	61.6	61.7	61.6
Sinequa	53.5	53.5	53.5	63.3	63.3	63.3	66.4	66.4	66.4
Talp	51.3	51.3	51.3	65.5	65.5	65.5	64.5	64.5	64.5
Pardon mfs	46.4	40.1	43.0	55.7	**53.6**	54.6	60.7	54.2	57.3
Pardon ufs	47.1	37.7	41.9	47.9	34.3	38.7	47.4	31.5	35.2

Table: Results in **P**recision and **R**ecall for each PoS

Discusion: What PARDON cannot do

- To overcome errors from the preprocessing steps (e.g in the PoS, the lemmatization or the tokenization, no parse tree).
- It is hard to disambiguate senses without previous examples.
- To disambiguate words whose syntactic behaviour does not vary and whose senses are similar (e.g. child).
- Only uses information from words which have a direct syntactic connection.

> ...
> As presented by Mr. Chabrol, and <**head**>*played*</**head**> with thin-lipped intensity by Isabelle Huppert, Marie-Louise (called Marie Latour in the **film**) was not a nice person.
> ...

Figure: Sentence example (play.131)

Outline

Outline

Conclusions

- NLU is a knowledge hungry complex task
- NLU systems must deal with inconsistences and incompleteness
- PARDON (Optimization techniques) is a feasible and scalable model to integrate:
  - processes
  - knowledge
- We have test PARDON in two well-known NLP tasks

Outline

Further Work

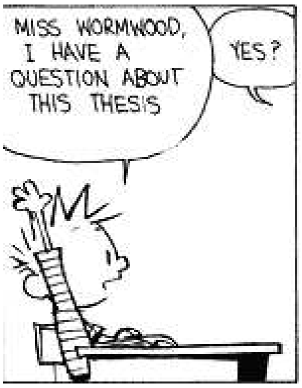
- Improve the similarity measure. (e.g. ML).
- The integration of multiple levels of NLP, e.g. PoS and Parsing, or Parsing and WSD.
- The application of PARDON to other NLP tasks: MWE detection, Parsing, etc.
- Integrating other knowledge sources (e.g. Wikipedia)
- Combine other WSD methods (e.g. Weighted classifiers [Escudero'06]WSD based on Domains [Magnini+Strapparava'2000]) with PARDON.
- Add anaphora resolution to break sentence boundaries.
- Explore unsupervised WSD and related issues, such as how to integrate supervised and non-supervised models or which models should be activated for a given word.

Thanks for your attention

Eskerrik asko zuen arretagatik
Gràcies per la vostre atenció

157

Questions?



158

[Appelt et al.'1996] D. Appelt, J. Hobbs, J. Bear, D. Israel amd M. Kameyama, A. Keheler, D. Martin, K. Myers, and M. Tyson.
1996.
SRI International FASTUS System:MUC-6 Test Results and Analysis.
In *Proceedings of the* 6^{th} *MUC Conference.*

[Cahill et al.'1999] Lynne Cahill, Evans R, Mellish C, Reape M, and Scott D.
1999.
Towards a Reference Architecture for Natural Language Generation Systems - The RAGS project.
Technical Report 2000 8/1999, Technical Report. Brighton: Information Technology Research Institute.

[Cahill et al.'2001] Lynne Cahill, John Carroll, Roger Evans, Daniel Paiva, Richard Power, Donia Scott, and Kees van Deemter.
2001.

From RAGS to RICHES: exploiting the potential of a flexible generation architecture.
In *Proceedings of ACL/EACL 2001*, pages 98–105, Toulouse, France.

[Decadt et al.'2004] Bart Decadt, Véronique Hoste, Walter Daelemans, and Antal van den Bosch.
2004.
GAMBL, Genetic Algorithm Optimization of Memory-Based WSD.
In R. Mihalcea and P. Edmonds, editors, *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-III)*, pages 108–112, Barcelona, Spain, July. Proceedings of ACL'2004.

[Grishman'1995] Ralph Grishman.
1995.
Nyu system or where's the syntax?
In *MUC-6*, pages 167–175.

[Hirst'1987] Graeme Hirst.

1987.
Semantic Interpretation and the Resolution of the ambiguity.
Studies in Natural Language Processing. Cambridge University Press.

[Hoste et al.'2001] Véronique Hoste, Anne Kool, and Walter Daelemans.
2001.
Classifier Optimization and Combination in the English All Words Task.
In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems (Senseval-II)*, pages 83–86, Toulouse, France.

[Magnini and Strapparava'2000] Bernardo Magnini and Carlo Strapparava.
2000.
Experiments in word domain disambiguation for parallel texts.
In *Proceedings of the ACL Workshop on Word Senses and Multilinguality*.

- [Ng and Lee'1996] Hwee Tou Ng and Hian Beng Lee.
  1996.
  Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach.
  In Arivind Joshi and Martha Palmer, editors, *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 40–47, San Francisco.
  Morgan Kaufmann Publishers.
- [Vázquez et al.'2000] Glória Vázquez, Ana Fernández, and Maria Antonia Martí.
  2000.
  Clasificación Verbal, Alternancias de Diátesis.
  Number 3 in Cuaderns de Sintagma. Sintagma.
- [Yangarber and Grishman'1998] R. Yangarber and R. Grishman.
  1998.
  NYU: Description of the Proteus/PET System as Used for MUC-7 ST.

MUS scores

The MUC evaluation metrics establish the following cases:

- **Correct** (*COR*): Roles correctly assigned by the system.
- **Incorrect** (*INC*): Roles incorrectly assigned by the system.
- **Missing** (*MIS*): Roles unassigned by the system when they should have been assigned.
- **Spurious** (*SPU*): Roles assigned by the system when they should have been unassigned.

These cases lead to the definition of the following measures, where **Possible** (POS) are the roles that should be assigned (COR+INC+MIS) and **Actual** (ACT) are the roles actually assigned by the system under evaluation (COR+INC+SPU):

- **Undergeneration** $UND = 100 \times \frac{MIS}{POS}$
- **Overgeneration** $OVR = 100 \times \frac{SPU}{ACT}$
- **Substitution** $SUB = 100 \times \frac{INC}{COR+INC}$
- **Error** $ERR = 100 \times \frac{INC+SPU+MIS}{COR+INC+SPU+MIS}$
- **Precision** $PRE = 100 \times \frac{COR}{ACT}$
- **Recall** $REC = 100 \times \frac{COR}{POS}$