Multilanguage Dependency Parsing

Giuseppe Attardi
Dipartimento di Informatica
Università di Pisa
Contributors: Massimo Ciaramita, Maria Simi

Language and Intelligence

"Understanding cannot be measured by external behavior; it is an internal metric of how the brain remembers things and uses its memories to make predictions".

"The difference between the intelligence of humans and other mammals is that we have language".

Jeff Hawkins, "On Intelligence", 2004

Hawkins' Memory-Prediction framework

- **The brain uses vast amounts of memory to create a model of the world. Everything you know and have learned is stored in this model. The brain uses this memory-based model to make continuous predictions of future events. It is the ability to make predictions about the future that is the crux of intelligence.**

More …

"Spoken and written words are just patterns in the world…

The syntax and semantics of language are not different from the hierarchical structure of everyday objects.

We associate spoken words with our memory of their physical and semantic counterparts.

Through language one human can invoke memories and create next juxtapositions of mental objects in another human."

Conclusion

- **Ability to process language will become essential in many computer applications**

Disruptive Technology

- **Heather McCallum-Bayliss, director at the Disruptive Technology Office**
- **Keynote at TREC 2006:**
  - **DTO is set to solve one hard problem: Human Language**

Is NLP needed?

- Many Information Retrieval tasks can do without
  - Document retrieval, categorization, summarization, information filtering
- Focus on Document Retrieval reduces the need for NLP techniques
  - Discourse factors can be ignored
  - Redundant words perform word-sense disambiguation
- Lack of robustness:
  - NLP techniques are typically not as robust as word indexing

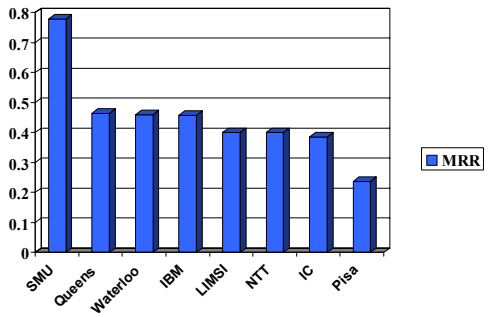
Semantic Document Analysis

- Question Answering
  - Return precise answer to natural language queries
- Relation Extraction
- Intent Mining
  - assess the attitude of the document author with respect to a given subject, e.g. *problem* (*description*, *solution*), *agreement* (*assent*, *dissent*), *preference* (*likes*, *dislikes*), *statement* (*claim*, *denial*)
  - Opinion mining: attitude is a positive or negative opinion

Question Answering is Different

- Search Engines return list of (possibly) *relevant* documents
- Users still to have to dig through returned list to find answer
- QA: give the user a (short) answer to their question, perhaps supported by evidence

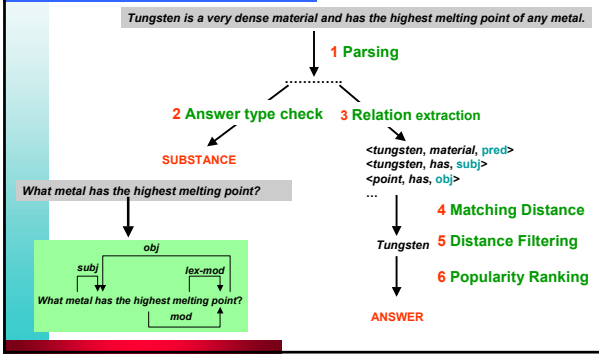
TREC 2000 Results (long)



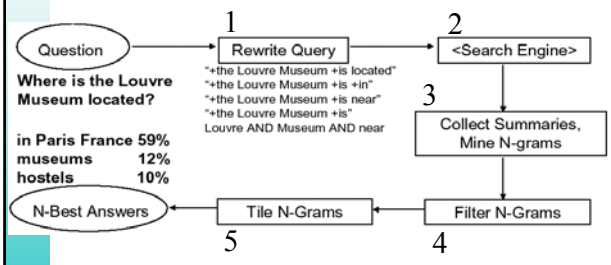
Falcon

- The Falcon system from SMU was by far best performing system at TREC 2000
- It used NLP and performed deep semantic processing

PiQASso Answer Matching



AskMSR: Web Mining



Step 1: Rewrite queries

- **Intuition: The user's question is often syntactically quite close to sentences that contain the answer**
  - **Where is the Louvre Museum located?**
  - **The Louvre Museum is located in *Paris***
  - **Who created the character of Scrooge?**
  - ***Charles Dickens* created the character of Scrooge.**

Query rewriting

- **Classify question into seven categories**
  - **Who is/was/are/were…?**
  - **When is/did/will/are/were …?**
  - **Where is/are/were …?**

a. Category-specific transformation rules

e.g. "For Where questions, move 'is' to all possible locations"

"Where is the Louvre Museum located"
- **→ "is the Louvre Museum located"**
- **→ "the is Louvre Museum located"**
- **→ "the Louvre is Museum located"**
- **→ "the Louvre Museum is located"**
- **→ "the Louvre Museum located is"**

Nonsense, but who cares? It's only a few more queries to Google.

b. Expected answer "Datatype" (eg, Date, Person, Location, …)

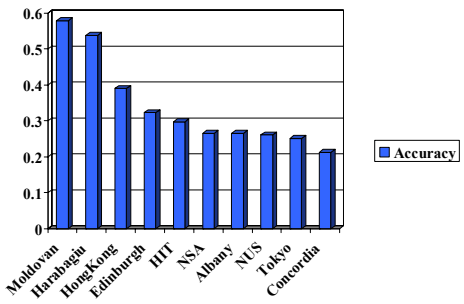
When was the French Revolution? → DATE

- **Hand-crafted classification/rewrite/datatype rules**

Step 2: Query search engine

- **Send all rewrites to a Web search engine**
- **Retrieve top N answers**
- **For speed, rely just on search engine's "snippets", not the full text of the actual document**

TREC 2006 QA: Factoid



Best QA Systems at TREC 2006

- **LCC PowerAnswer (58% accuracy), LCC Chaucer (54%)**
- **Both use NLP tools to tokenize, POS tag and syntactic parse each question**

Chaucer: Answer Extraction

- **Strategies:**
  1. **NE**
     - **select passages containing answer NE**
     - **finds quite high number of correct answers**
  2. **hard pattern-based (hand-crafted rules)**
  3. **authoritative sources (retrieved from Web: imdb)**
  4. **soft pattern based (Cui, Kan, & Chua 2004)**
  5. **FrameNet: answer parsed and ranked according to semantic frames**
  6. **predictive question-based**

FrameNet Matching example

- ***Question*: How old was Padre Pio when he died?**
  - **Frames: Age[Padre Pio], Death[Padre Pio]**
- ***Answer*: Padre Pio, who died in 1968 at the age of 81, was right.**
  - **Frames: Age[Padre Pio, 81], Death[Padre Pio, 1968]**

Answer Selection

- **Uses Textual Entailment system**
- **Less formal that logical entailment. Satisfied is one passage can be reasonably inferred.**

NL Parser

Parser Requirements

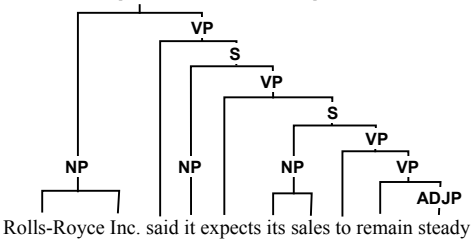
- **Multilanguage**
  - **Trainable on annotated corpora**
- **Accurate**
  - **Close to state of the art**
- **Flexible**
  - **Retrainable through unannotated corpora**
- **Efficient**
  - **Hundreds sentence/sec**
  - **Deterministic bottom-up parser**

Statistical Parsers

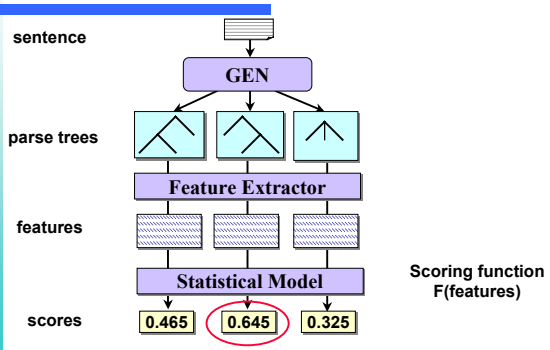
- **Major breakthrough in computational linguistics in recent years**
- **Parser types:**
  - **Constituent parser**
  - **Dependency parser**

Constituent Parsing

- Requires Phrase Structure Grammar
  - CFG, PCFG, Unification Grammar
- Produces phrase structure parse tree



Linear Parsing Model



Statistical Classifier

- Given a set of train examples, learn weights for each features, in order to compute scoring function

Constituent Parsing

- *GEN*: e.g. CFG (Context-Free Grammar)
- $h_i(x)$ feature of tree x, based on aspects of the tree

e.g.

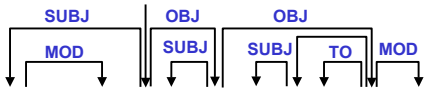
$h(x)$ = # of times  occurs in x

Constituent Parsers

- Probabilistic Generative Model of Language which include parse structure (e.g. Collins 1997)
- Conditional parsing models (Charniak 2000; McDonald 2005)

Dependency Parsing

- Produces dependency trees
- Word-word dependency relations
- Far easier to understand and to annotate



Generative Dependency Parsing

- McDonald et al. 2005
- *GEN* generates all possible Spanning Trees
- Select Maximum ST with best feature vector

Shift/Reduce Dependency Parser

- Traditional statistical parsers are trained directly on the task of selecting a parse tree for a sentence
- Instead a Shift/Reduce parser is trained and learns the sequence of parse actions required to build the parse tree

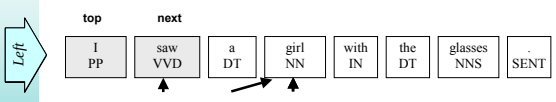
Grammar Not Required

- A traditional parser requires a grammar for generating candidate trees
- A Shift/Reduce parser needs no such grammar

Parsing as Classification

- Parsing based on Shift/Reduce actions
- Learns from annotated corpus which action to perform at each step
- Proposed by (Yamada-Matsumoto 2003) and (Nivre 2003)
- Uses only local information, but it can exploit history

Parser Actions



Dependency Graph

$D = \{d_1, \ldots, d_m\}$ set of dependency types

A dependency graph for a sequence of words $W = w_1 \ldots w_n$ is a labeled directed acyclic graph

$G = (W, A)$, where

(a) W is the set of nodes

(b) A is a set of labeled arcs (w_i, d, w_j)
$w_i, w_j \in W, d \in D$

(c) $\forall w_j \in W$, there is at most one arc
$(w_i, d, w_j) \in A$

Parser State

The parser state is a quadruple $\langle S, I, T, A \rangle$, where

- S **is a stack of partially processed tokens**
- I **is a list of (remaining) input tokens**
- T **is a stack of temporary tokens**
- A **is the arc relation for the dependency graph**

$(w, r, h) \in A$ represents an arc $w \rightarrow h$, tagged with dependency r

Parser Actions

Shift

$$\frac{\langle S, n|I, T, A \rangle}{\langle n|S, I, T, A \rangle}$$

Right

$$\frac{\langle s|S, n|I, T, A \rangle}{\langle S, n|I, T, A \cup \{(s, r, n)\} \rangle}$$

Left

$$\frac{\langle s|S, n|I, T, A \rangle}{\langle S, s|I, T, A \cup \{(n, r, s)\} \rangle}$$

Parser Algorithm

- **The parsing algorithm is fully deterministic:**

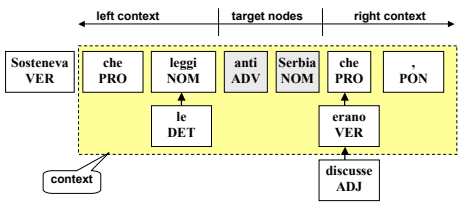
```
Input Sentence: (w1, p1), (w2, p2), … , (wn, pn)
  S = <>
  I = <(w1, p1), (w2, p2), … , (wn, pn)>
  T = <>
  A = { }
  while I ≠ <> do begin
    x = getContext(S, I, T, A);
    y = estimateAction(model, x);
    performAction(y, S, I, T, A);
  end
```

Learning Phase

Learning Features

feature	Value
W	word
L	lemma
P	part of speech (POS) tag
M	morphology: e.g. singular/plural
W<	word of the leftmost child node
L<	lemma of the leftmost child node
P<	POS tag of the leftmost child node, if present
M<	whether the rightmost child node is singular/plural
W>	word of the rightmost child node
L>	lemma of the rightmost child node
P>	POS tag of the rightmost child node, if present
M>	whether the rightmost child node is singular/plural

Learning Event



(-3, W, che), (-3, P, PRO),
(-2, W, leggi), (-2, P, NOM), (-2, M, P), (-2, W<, le), (-2, P<, DET), (-2, M<, P),
(-1, W, anti), (-1, P, ADV),
(0, W, Serbia), (0, P, NOM), (0, M, S),
(+1, W, che), (+1, P, PRO), (+1, W>, erano), (+1, P>, VER), (+1, M>, P),
(+2, W, ,), (+2, P, PON)

DeSR Parser Architecture

- **Modular learners architecture:**
  - MaxEntropy, MBL, SVM, Winnow, Perceptron
- **Classifier combinations: e.g. multiple MEs, SVM + ME**
- **Features can be selected**

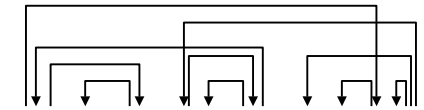
Feature used in Experiments

LemmaFeatures	-2 -1 0 1 2 3
PosFeatures	-2 -1 0 1 2 3
MorphoFeatures	-1 0 1 2
PosLeftChildren	2
PosLeftChild	-1 0
DepLeftChild	-1 0
PosRightChildren	2
PosRightChild	-1 0
DepRightChild	-1
PastActions	1

Projectivity

- **An arc $w_i \rightarrow w_k$ is projective iff**
  $\forall j, i < j < k$ or $i > j > k$,
  $w_i \rightarrow^* w_k$
- **A dependency tree is projective iff every arc is projective**
- **Intuitively: arcs can be drawn on a plane without intersections**

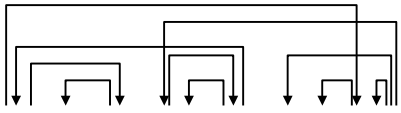
Non Projective



Actions for non-projective arcs

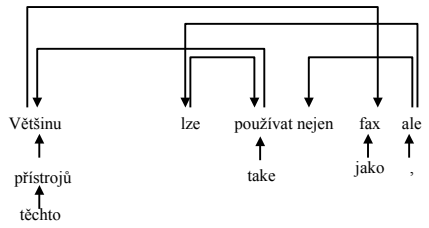
Right2	$\frac{\langle s_1 \mid s_2 \mid S, n \mid I, T, A \rangle}{\langle s_1 \mid S, n \mid I, T, A \cup \{(s_2, r, n)\} \rangle}$
Left2	$\frac{\langle s_1 \mid s_2 \mid S, n \mid I, T, A \rangle}{\langle s_2 \mid S, s_1 \mid I, T, A \cup \{(n, r, s_2)\} \rangle}$
Right3	$\frac{\langle s_1 \mid s_2 \mid s_3 \mid S, n \mid I, T, A \rangle}{\langle s_1 \mid s_2 \mid S, n \mid I, T, A \cup \{(s_3, r, n)\} \rangle}$
Left3	$\frac{\langle s_1 \mid s_2 \mid s_3 \mid S, n \mid I, T, A \rangle}{\langle s_2 \mid s_3 \mid S, s_1 \mid I, T, A \cup \{(n, r, s_3)\} \rangle}$
Extract	$\frac{\langle s_1 \mid s_2 \mid S, n \mid I, T, A \rangle}{\langle n \mid s_1 \mid S, I, s_2 \mid T, A \rangle}$
Insert	$\frac{\langle S, I, s_1 \mid T, A \rangle}{\langle s_1 \mid S, I, T, A \rangle}$

Example

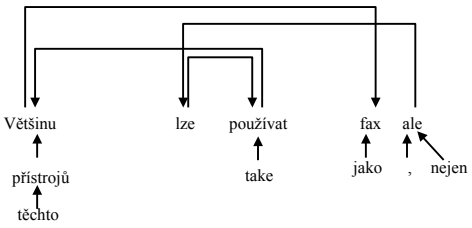


- ***Right2* (*nejen* → *ale*) and *Left3* (*fax* → *Většinu*)**

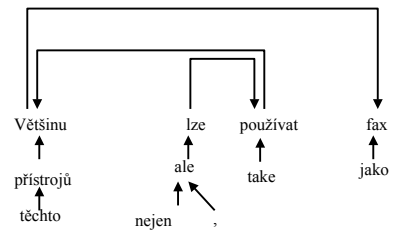
Example



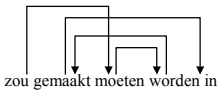
Example



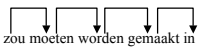
Example



Examples



Extract followed by *Insert*



Effectiveness for Non-Projectivity

- **Training data for Czech contains 28081 non-projective relations**
- **26346 (93%) can be handled by Left2/Right2**
- **1683 (6%) by Left3/Right3**
- **52 (0.2%) require Extract/Insert**

CoNLL-X Shared Task

- **To assign labeled dependency structures for a range of languages by means of a fully automatic dependency parser**
- **Input: tokenized and tagged sentences**
- **Tags: token, lemma, POS, morpho features, ref. to head, dependency label**
- **For each token, the parser must output its head and the corresponding dependency relation**

CoNLL-X: Collections

	Ar	Cn	Cz	Dk	Du	De	Jp	Pt	Sl	Sp	Se	Tr	Bu
K tokens	54	337	1,249	94	195	700	151	207	29	89	191	58	190
K sents	1.5	57.0	72.7	5.2	13.3	39.2	17.0	9.1	1.5	3.3	11.0	5.0	12.8
Tokens/sentence	37.2	5.9	17.2	18.2	14.6	17.8	8.9	22.8	18.7	27.0	17.3	11.5	14.8
CPOSTAG	14	22	12	10	13	52	20	15	11	15	37	14	11
POSTAG	19	303	63	24	302	52	77	21	28	38	37	30	53
FEATS	19	0	61	47	81	0	4	146	51	33	0	82	50
DEPREL	27	82	78	52	26	46	7	55	25	21	56	25	18
% non-project. relations	0.4	0.0	1.9	1.0	5.4	2.3	1.1	1.3	1.9	0.1	1.0	1.5	0.4
% non-project. sentences	11.2	0.0	23.2	15.6	36.4	27.8	5.3	18.9	22.2	1.7	9.8	11.6	5.4

CoNLL: Evaluation Metrics

- **Labeled Attachment Score (LAS)**
  - proportion of "scoring" tokens that are assigned both the correct head and the correct dependency relation label
- **Unlabeled Attachment Score (UAS)**
  - proportion of "scoring" tokens that are assigned the correct head

Shared Task Unofficial Results

Language	Maximum Entropy				MBL			
	LAS %	UAS %	Train sec	Parse sec	LAS %	UAS %	Train sec	Parse sec
Arabic	56.43	**70.96**	181	2.6	59.70	**74.69**	24	950
Bulgarian	82.88	**87.39**	452	1.5	79.17	**85.92**	88	353
Chinese	81.69	**86.76**	1,156	1.8	72.17	**83.08**	540	478
Czech	62.10	**73.44**	13,800	12.8	69.20	**80.22**	496	13,500
Danish	77.49	**83.03**	386	3.2	78.46	**85.21**	52	627
Dutch	70.49	**74.99**	679	3.3	72.47	**77.61**	132	923
Japanese	84.17	**87.15**	129	0.8	85.19	**87.79**	44	97
German	80.01	**83.37**	9,315	4.3	79.79	**84.31**	1,399	3,756
Portuguese	79.40	**87.70**	1,044	4.9	80.97	**87.74**	160	670
Slovene	61.97	**74.78**	98	3.0	62.67	**76.60**	16	547
Spanish	72.35	**76.06**	204	2.4	74.37	**79.70**	54	769
Swedish	78.35	**84.68**	1,424	2.9	74.85	**83.73**	96	1,177
Turkish	58.81	**69.79**	177	2.3	47.58	**65.25**	43	727

Latest Improvements

Language	UAS	LAS
English	91.00	92.03
Swedish	83.87	89.40
Spanish	80.16	83.51
Slovene	70.52	80.50

CoNLL-X: Comparative Results

	LAS		UAS	
	Average	Ours	Average	Ours
Arabic	**59.94**	59.70	73.48	**74.69**
Bulgarian	79.98	**82.88**	85.89	**87.39**
Chinese	78.32	**81.69**	84.85	**86.76**
Czech	67.17	**69.20**	77.01	**80.22**
Danish	78.31	**78.46**	84.52	**85.21**
Dutch	70.73	**72.47**	75.07	**77.71**
Japanese	**85.86**	85.19	**89.05**	87.79
German	78.58	**80.01**	82.60	**84.31**
Portuguese	80.63	**80.97**	86.46	**87.74**
Slovene	**65.16**	62.67	76.53	**76.60**
Spanish	73.52	**74.37**	77.76	**79.70**
Swedish	76.44	**78.35**	84.21	**84.68**
Turkish	55.95	**58.81**	69.35	**69.79**

Average scores from 36 participant submissions

Performance Comparison

- **Running an SVM-based parser (Maltparser 0.4) on same Xeon 2.8 MHz machine**
- **Training on Swedish Talbanken:**
  - **390 min**
- **Test on CoNLL Swedish test set (~5000 tokens):**
  - **13 min**

Why performance

- It can be run on massive collection
  - Text mining
  - Semantic Document Analysis (news)
  - Semantic Web
- It can be run on your desktop:
  - Desktop search
  - Mail analysis (e.g. better spam)
  - Semantic document correlation

Revising Parse Trees

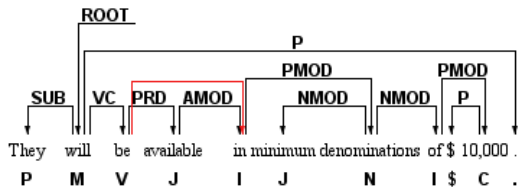
Reranking Approaches

- Base parser produces a set of candidate parses, with initial associated
- A second statistical model is trained on the output trees, using a different set of features
- used to rerank these parses assigning a score to each
- Reranking achieves error reduction up to 13% (Collins and Koo 2006)

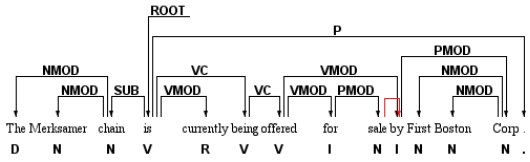
Intuition

- A dependency parser is mostly right in identifying chunks
- Only local corrections are needed to rearrange such chunks
- Generating several parses would often just repeat the same steps

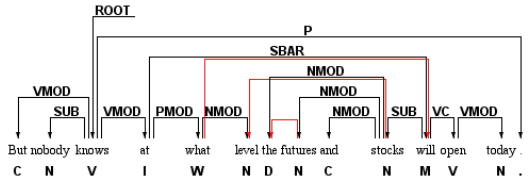
Sample Error



Sample errors



Sample Error



Revision Learning

- Learn tree transformations to correct errors
- Additional information can be gathered from trees
- Including semantic information
  - Named Entity
  - Semantic Role Label

What Kind of Tree Transformations?

- Simple to identify
- Simple to apply
- Simplification assumptions:
  - No node addition/deletion
  - Transformations are Independent

Revision Rules

Step	Description
u	Up to parent
-1	Left to the *n*-th token
+1	Right to the *n*-th token
[	Head of previous constituent
]	Head of following constituent
>	First token of previous constituent
<	First token of following constituent
d--	Down to leftmost child
d++	Down to rightmost child
d-1	Down to first left child
d+1	Down to first right child
dP	Down to token with POS P

Top 15 Revision Rules

Freq	Rule	Target node
983	uu	Up, up
685	-1	Token to the left
469	+1	Token to the right
265	[	Head of previous constituent
215	uuu	Up, up, up
197	+1u	One right, up
194	r	To root
174	-1u	One left, up
116	>u	To token after constituent, then up
103	ud--	Up down to leftmost child
90	dV	To first child token with POS verb
83	d+1	Down to first right child
82	uuuu	Up, up, up, up
74	<	To token before constituent
73	ud+1	Up down to first right child

Revision Problem

Given $G = (s, A)$ for sentence $s = \langle w_1 \ldots w_n \rangle$

A revision rule is a mapping $r_i : A \rightarrow A$, such that $r_i(w_i, d, w_j) = (w_i, d', w_j')$

Compute $R = \langle r1, \ldots, rn \rangle$

Transformed tree: $R(G) = (s, A')$ where $A' = \{ r_i(w_i, d, w_j) \mid (w_i, d, w^j) \in A \}$

Results: WSJ

Parser	UAS	LAS
DeSR-ME	84.96	83.53
DeSR-MBL	88.41	86.85
DeSR-MBL-Revision	89.11	86.39
DeSR-ME-Revision	90.27	Linear 4
McDonald	91.50	Cubic –
Nivre&Scholz	87.3	–
Y&M	90.3	–

Remarks

- Somewhat surprisingly, training for revisions on the data produced by the more accurate parser gives lower error reduction
- Suggestion: generate bad parser on purpose
- For instance: use less training data

Results: Swedish

Parser	UAS	LAS
DeSR-SVM	88.41	83.31
DeSR-SVM-Revision	89.76	83.13
Corston-Oliver	89.54	82.33
Nivre	89.50	84.58

Remarks

- Tree Revision:
  - Simple idea
  - Effective
  - Can still be improved

TreeBank Annotations

TreeBank for Parsing: Constraints

- all tokens from the original sentence must be present, including punctuations
- no extra tokens must be present
- each token must have exactly one dependency link
- dependency links make a tree without cycles, with a single root

Adapting TreeBank to Parsing

- **TreeBanks can be designed for different goals**
  - **Linguists: representing accurately all linguistic phenomena**
  - **Computational linguists:**
    - **statistical analysis**
    - **input for machine-learning tools**
  - **Semanticists:**
    - **analyzers of semantics and discourse structure**

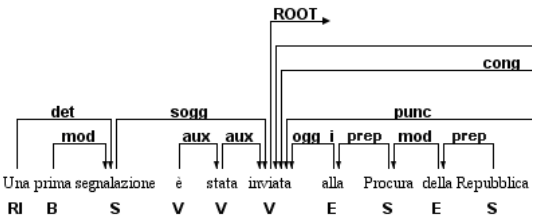
Dependency Structures

- **Dependency formalism is non prescriptive**
- **Different ways to represent same meaning**
- **For example:**
  - **Coordination (“apples, peaches and pears”)**
  - **Preposition-noun relation**
  - **Finite verb to auxiliary relation (“has gone”)**

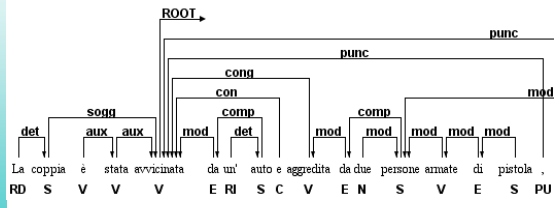
Parser Accuracy

- **Nivre experiments on Swedish Talbanken:**
  - **conjuncts will together with the conjunctions form a chain**
  - **Preposition-noun relation: preposition becomes the head**
  - **finite verb is the head of the non-finite verb**
- **Improvement of 2.4% in UAS**

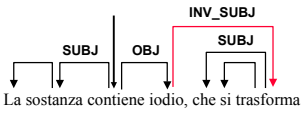
Example: aux



Example: conjunctions



Example: relatives



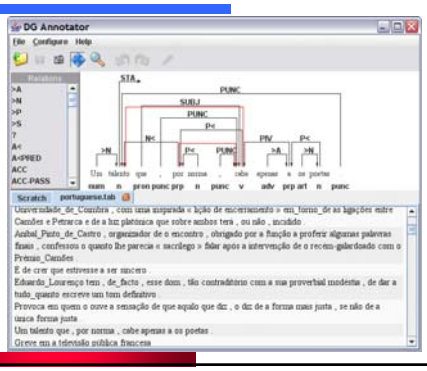
Italian Treebank

- Cooperation with CNR ILC
- Access to SI-TAL corpus
- Working on completing annotation and converting to CoNLL format
- Semiautomated process: heuristics + manual fixup

DgAnnotator

- A GUI tool for:
  - Annotating texts with dependency relations
  - Visualizing and comparing trees
  - Generating corpora in XML or CoNLL format
  - Exporting DG trees to PNG
- Demo
- Available at: http://medialab.di.unipi.it/Project/QA/Parser/DgAnnotator/

DgAnnotator Example



Dislocation phenomena

- A María Juan *la* vio ayer.
- A tu hermano Juan no *lo* puede ni ver.
- Ese libro el nio debe leer*lo* cuanto antes.
- A Pedro la carta hay que escribír*sela* pronto.
- I dolci *li* ho mangiati ieri.
- Di questo non *ne* voglio parlare.
- A Roma io non *ci* vado.
- Al jardí els nens s'*hi* diveteixen molt.

Opinion Mining

TREC Blog 2006

- Resources
- Collection of blogs
  - from a crawl of 100,649 RSS/Atom feeds, over 11 weeks
  - Several languages
  - Includes spam
- Set of 4000 opinionated words, with both positive/negative weights
  - Computed by [Esuli & Sebastiani, 2005] using classifier trained on WordNet glosses

Sample Topics

- Larry Summers
- Macbook pro
- Skype 2.0
- Colbert Report

Opinion Retrieval

- Typical 2-stage approach:
  1. Perform IR and rank by topic relevance
  2. Postprocess results with filters and rerank
- Single-stage approach:
  - Enrich the index with opinion tags
  - Perform normal retrieval with custom ranking function

Enriched Index

- Overlay words with tags

1	2	3	4	5
music	is	a	touch	lame
				NEGATIVE
soundtrack			little	weak
ART			bit	plate

Enhanced Queries

- music NEGATIVE:lame
- music NEGATIVE:*

Opinionated proximity

Query: proximity 3 [OPINIONATED:* skype]

- Skype Doesn't Want My Money
- … guys working on Skype and no one really took it seriously
- Para aquellos que no conozcan Skype

Accuracy-Performance Tradeoff

run	time	p@5
University of Chicago	12 hours	52.00
University of Amsterdam		48.80
University of Pisa	6.28 sec	47.60

References

- G. Attardi. 2006. Experiments with a Multilanguage Non-projective Dependency Parser. In Proc. CoNLL-X.
- H. Yamada, Y. Matsumoto. 2003. Statistical Dependency Analysis with Support Vector Machines. In *Proc. of IWPT-2003*.
- J. Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proc. of IWPT-2003*, pages 149–160.
- A. Esuli, F. Sebastiani. 2005. Determining the semantic orientation of terms through gloss analysis. In Proc. of CIKM-05.