Parsing the *Index Thomisticus* Treebank

Some preliminary results

Marco Passarotti   Paolo Ruffolo

GIRCSE
Catholic University of the Sacred Heart
Milano

XV Colloquium Internationale Linguisticae Latinae, Innsbruck, 2009

Outline

Outline

Dependency Grammar

DG Tree (Tesnière, Mel'cuk, etc.):

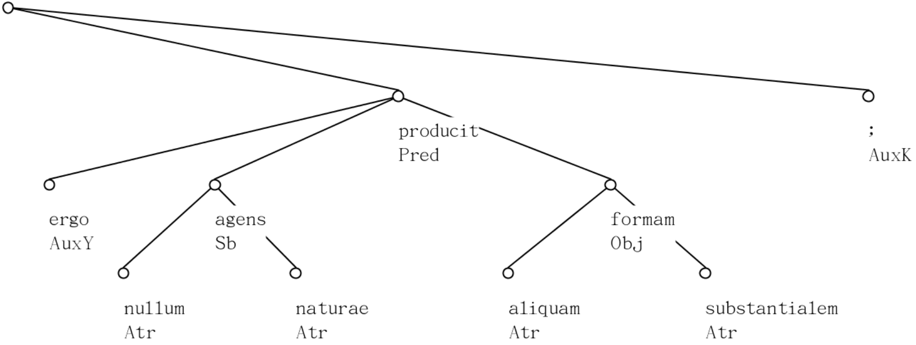
- Only lexical nodes (Chomsky's terminals).
- Lexical nodes are linked through binary asymmetric relations ('dependencies') and are tagged with functional labels.
- Sentence word order not marked.
- Suitable for "free-word-order" languages like Latin.

DG example

```
ergo nullum agens naturae producit aliquam formam substantialem;
```

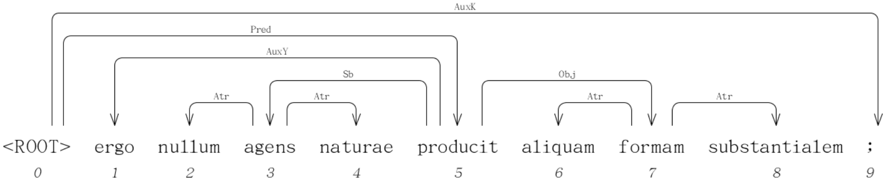
DG example

ergo nullum agens naturae producit aliquam formam substantialem;



DG example

ergo nullum agens naturae producit aliquam formam substantialem;



Outline

Probabilistic Dependency Parsers

Train Mode

INPUT Set Of Dependency Tagged Sentences

OUTPUT Model

Parse Mode

INPUTS Sentence(s), Model

OUTPUT Dependency Tagged Sentence(s)

Algorithms Classification

Shift-Reduce

Processing Each token sentence is processed in a linear order.

Data Structure A stack is used to store tokens being processed.

Method A local optimization criterion is adopted to best fit data.

Cost Generally linear time cost.

Graph-Based

Processing Each sentence is processed as a whole.

Data Structure A support graph is used.

Method A global optimization criterion is adopted to best fit data.

Cost Generally quadratic time cost.

The Parsers We Used

Shift-Reduce

- DeSR (Attardi et al.)
- MaltParser (Nivre et al.)
- ISBN (Titov and Henderson)

Graph-Based

- MST (McDonald et al.)

Outline

Sources and Sampling Description

IT-TB : SGML-CSTS encoded data of the *Index Thomisticus* Treebank.

- PoS tagged, Morphological information is provided.
- Data belong exclusively to sentences containing at least one occurrence of lemma *forma*.

LDT : XML encoded data of the Latin Dependency Treebank.

- PoS tagged, Morphological information is provided.

Training Sets and Test Sets

IT-TB and LDT data sets were randomly partitioned on a ratio of 1:9

Training Sets : The 9/10 of IT-TB and LDT sentences were used to train the parsers

Test Sets : The remaining 1/10 was used to test the parsing process

Data Sets Description

Data Sets	Sentences	Tokens
IT-Train	2007	44195
IT-Test	243	5697
LDT-Train	3093	47662
LDT-Test	380	5481

Non-Projectivity

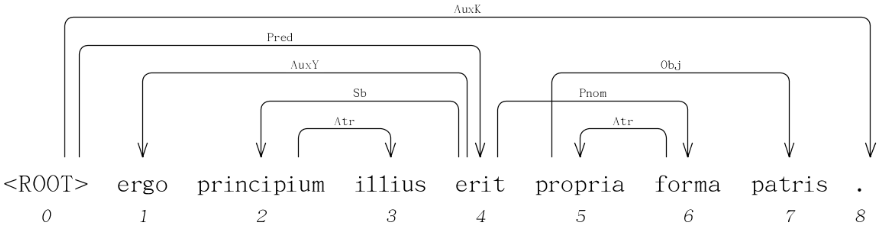
Projectivity

A dependency structure is said projective if for all tokens between the head and its dependent node (in left-right order), there can only be direct or indirect dependence of the head.

Non-Projectivity

Projectivity

A dependency structure is said projective if for all tokens between the head and its dependent node (in left-right order), there can only be direct or indirect dependence of the head.



Non-Projectivity In Data Sets

Data Sets	Tokens	Non-Projective	Rate
IT-Train	44195	1435	3,25%
IT-Test	5697	181	3,18%
IT	49892	1616	3,24%
LDT-Train	47662	3194	6,70%
LDT-Test	5481	339	6,19%
LDT	53143	3533	6,65%

Outline

Evaluation Metrics

CoNLL Shared Task 2006
(Buchholz and Marsi, 2006)

LAS Labeled Attachment Score, the percentage of tokens with correct head and relation label

UAS Unlabeled Attachment Score, the percentage of tokens with correct head

LA Label Accuracy, the percentage of tokens with correct relation label

Best Accuracy Results for Each Parser

Best Parsing Results

Parser	LAS	UAS	LA
DeSR	71,26%	78,35%	81,07%
Malt	69,85%	75,87%	81,74%
MST	68,79%	79,43%	79,35%
ISBN	68,97%	77,79%	78,88%

Tuning

Data Sets Variation Experiments

We applied to DeSR combined training sets and test sets:

- LDT samples were parsed by DeSR trained on IT-TB train set.
- IT-TB test set was parsed by DeSR trained on LDT train set.
- IT-TB and LDT test sets were parsed by DeSR trained on the set resulting from joining IT-TB and LDT training sets.

Results from Crossing Data Sets

IT-Train

Test Set	LAS	UAS	LA
Caesar	10,70%	18,22%	14,53%
Cicero	12,12%	18,80%	16,15%
Jerome	12,91%	19,32%	14,96%
Ovid	9,16%	18,04%	14,10%
Petronius	12,96%	24,73%	15,57%
Propertius	8,07%	17,91%	13,07%
Sallustius	11,01%	19,80%	14,83%
Vergil	9,43%	19,39%	12,42%

Compare

Results from Crossing Data Sets

LDT-Train

Test Set	LAS	UAS	LA
IT-Test	12,98%	30,74%	19,30%

Compare

Results from Mixing Data Sets

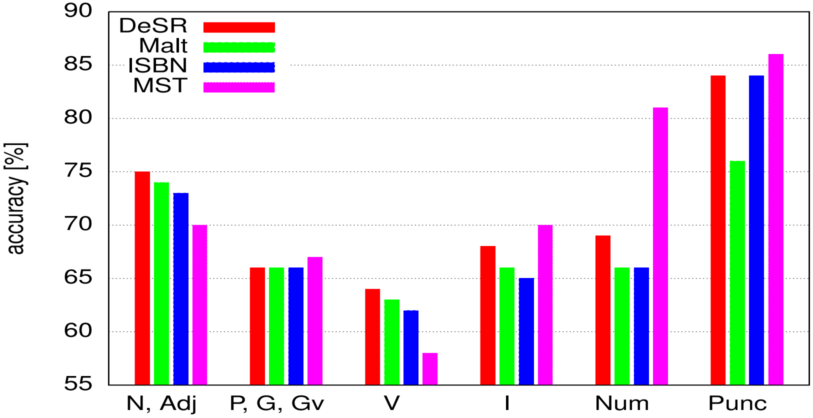
IT-Train + LDT-Train

Test Set	LAS	UAS	LA
LDT-Test	50,44%	59,52%	63,78%
IT-Test	71,82%	78,59%	81,89%

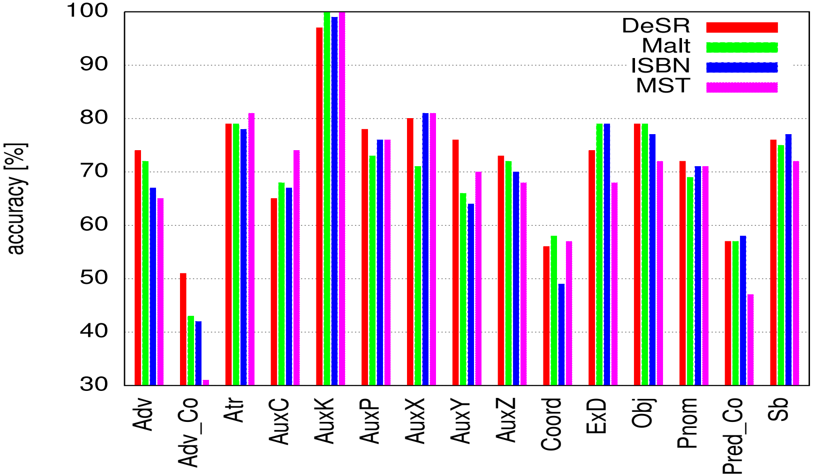
Compare

Outline

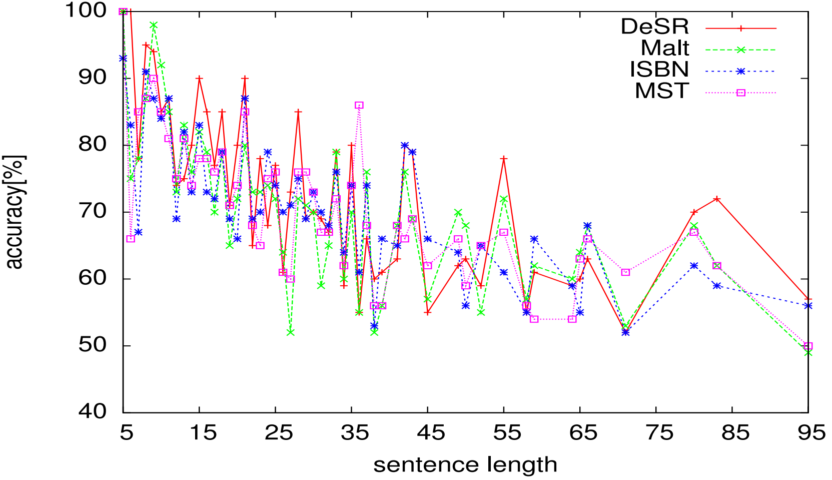
Accuracy By PoS



Accuracy By Dependency Relation

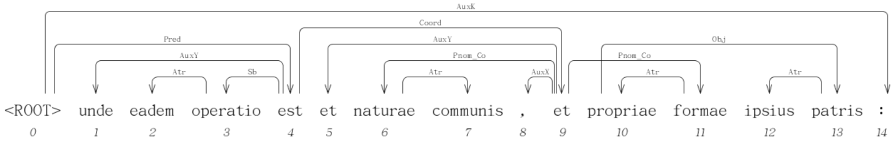


Accuracy By Sentence Length



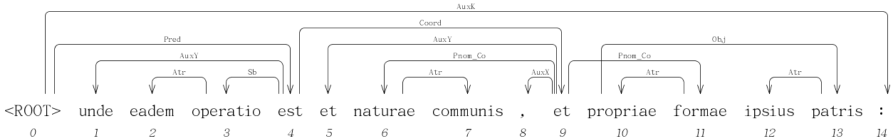
Visual Analysis

Gold-Standard:

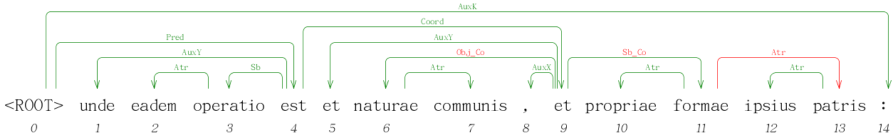


Visual Analysis

Gold-Standard:

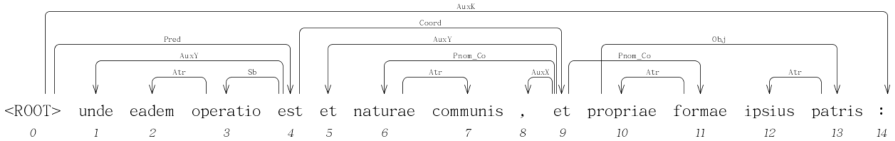


DeSR-Parsed:

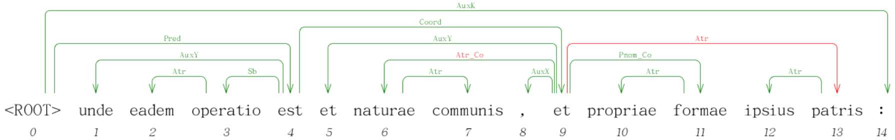


Visual Analysis

Gold-Standard:

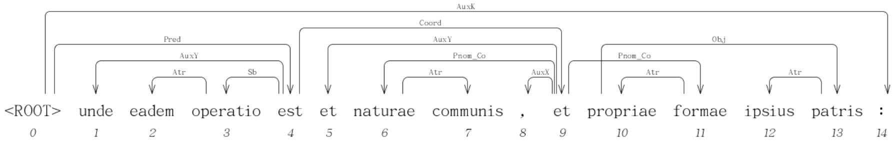


Malt-Parsed:

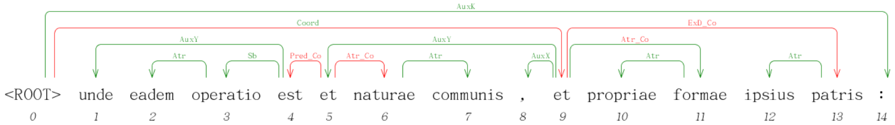


Visual Analysis

Gold-Standard:

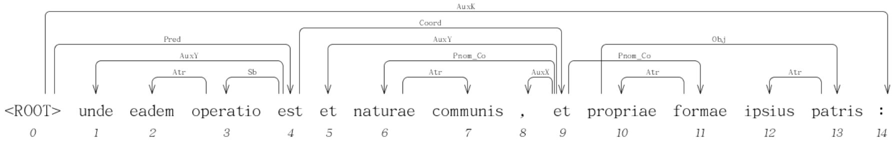


MST-Parsed:

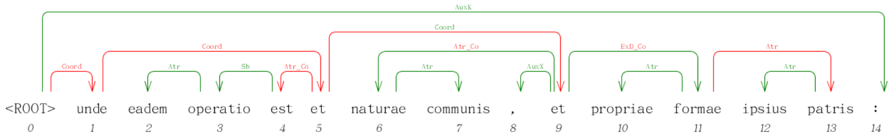


Visual Analysis

Gold-Standard:



ISBN-Parsed:



Conclusion and Perspectives

- Trained parsers are able to improve annotators performances in terms of time and quality.
- Bigger and better data sets can, in turn, improve parser accuracy.
- ...a virtuous circle.

- Outlook
  - Perform an in-depth error analysis.
  - Build up a features set tailored for Latin.
  - IT-TB PoS Tagging.
  - Explore other parser techniques: Parser Combination.
  - Find a way to combine data from several Latin treebanks.

DG CSTS annotation example

```
ergo nullum agens naturae producit aliquam formam substantialem;
```

```
<s id="002.2SN.DS-1QU1.AR4-AG-5.8-5.9-3">
<f>ergo<l>ergo<t>4-O-----<A>AuxY<r>1<g>5
<f>nullum<l>nullus<t>11F--A3--<A>Atr<r>2<g>3
<f>agens<l>agens<t>11C--A3--<A>Sb<r>3<g>5
<f>naturae<l>natura<t>11A--B2-1<A>Atr<r>4<g>3
<f>producit<l>produco<t>3-NA1-6--<A>Pred<r>5<g>0
<f>aliquam<l>aliqui<t>11F--D2--<A>Atr<r>6<g>7
<f>formam<l>forma<t>11A--D2--<A>Obj<r>7<g>5
<f>substantialem<l>substantialis<t>11C--D2--<A>Atr<r>8<g>7
<d>;<l>;<t>-------<A>AuxK<r>9<g>0
```

LDT Data Description

Data Sets	Sentences	Tokens
Caesar	71	1488
Cicero	327	6229
Jerome	405	8382
Ovid	316	4789
Petronius	1114	12474
Propertius	361	4857
Sallustius	701	12311
Vergil	178	2613

Parsers Tuning

Features Selection

- English
- Czech
- Italian

Non-Projectivity behavior

- All the parser except DeSR allow specific setting for non-projectivity.
- MaltParser allows the selection of several 'projectivezing' algorithms.

Parsers Tuning

Algorithms

- DeSR can be set in order to use several learning algorithms.
- MaltParser offers the possibility to select the parsing algorithm among three (Covington, Nivre Standard, Nivre Arc-eager).

Parser Tuning: Features Selection (MaltParser)

Features	LAS	UAS	LA
default	63,79%	71,24%	72,72%
czech	69,24%	75,56%	77,18%
italian	69,63%	75,36%	77,49%

Parser Tuning: Projectivity (MaltParser)

Algorithm	**LAS**	**UAS**	**LA**
Baseline	69,85%	75,87%	81,74%
Head	69,83%	75,67%	80,88%
Path	69,67%	75,44%	81,03%
Head+Path	69,18%	75,09%	80,80%

Parser Tuning: Algorithms

Parser	Algorithm	LAS	UAS	LA
Malt	Covington	64,72%	72,00%	74,76%
Malt	Nivre arceager	63,79%	71,24%	72,72%
Malt	Nivre arcstandard	57,90%	68,73%	65,19%
MST	non projective	68,79%	79,43%	79,35%
MST	projective	67,15%	78,12%	78,53%

IT-TB and LDT Parsed By DeSR

Straight Parsing

Train Set	Test Set	LAS	UAS	LA
LDT-Train	LDT-Test	51,18%	60,28%	63,67%
IT-Train	IT-Test	71,26%	78,35%	81,07%

IT-TB and LDT Parsed By DeSR

Straight Parsing

Train Set	Test Set	LAS	UAS	LA
LDT-Train	LDT-Test	51,18%	60,28%	63,67%
IT-Train	IT-Test	71,26%	78,35%	81,07%

IT-TB and LDT Parsed By DeSR

Straight Parsing

Train Set	Test Set	LAS	UAS	LA
LDT-Train	LDT-Test	51,18%	60,28%	63,67%
IT-Train	IT-Test	71,26%	78,35%	81,07%

Accuracy By PoS

Cpostag	DeSR	Malt	ISBN	MST
N, Adj	75,92%	74,36%	73,72%	70,96%
P, G, Gv	66,48%	66,48%	66,48%	67,04%
V	64,63%	63,54%	62,18%	58,50%
I	68,17%	66,69%	65,57%	70,27%
Num	69,70%	66,67%	66,67%	81,82%
Punc	84,46%	76,45%	84,96%	86,56%

Accuracy By Dependency Relations

Deprel	DeSR	Malt	ISBN	MST
Adv	74,78%	72,63%	67,93%	65,32%
Adv_Co	51,02%	43,56%	42,62%	31,96%
Atr	79,59%	79,66%	78,36%	81,75%
AuxC	65,00%	68,35%	67,73%	74,40%
AuxK	97,46%	100,00%	99,57%	100,00%
AuxP	78,68%	73,86%	76,48%	76,99%
AuxX	80,71%	71,74%	81,21%	81,32%
AuxY	76,92%	66,44%	64,23%	70,59%
AuxZ	73,54%	72,45%	70,97%	68,18%
Coord	56,51%	58,43%	49,38%	57,88%
ExD	74,57%	79,59%	79,17%	68,45%
Obj	79,11%	79,06%	77,64%	72,79%
Pnom	72,79%	69,13%	71,13%	71,71%
Pred_Co	57,94%	57,48%	58,33%	47,74%
Sb	76,26%	75,93%	77,87%	72,33%