UPF at EPE 2017: Transduction-based Deep Analysis

Simon Mille[1], Roberto Carlini[1], Ivan Latorre[1], Leo Wanner[1,2]

[1]Pompeu Fabra University, Barcelona

[2]ICREA , Barcelona

Pisa – Sept 20th 2017

Introduction

Separate levels of representation

- NLG: [Tesnière, 1959; Rambow & Korelsky, 1992; Reiter & Dale, 1997; Mellish *et al*, 2006, Mel'čuk since he was 2 y.o., etc.]
- Prague school, etc.

Three runs submitted

- Surface-syntactic output

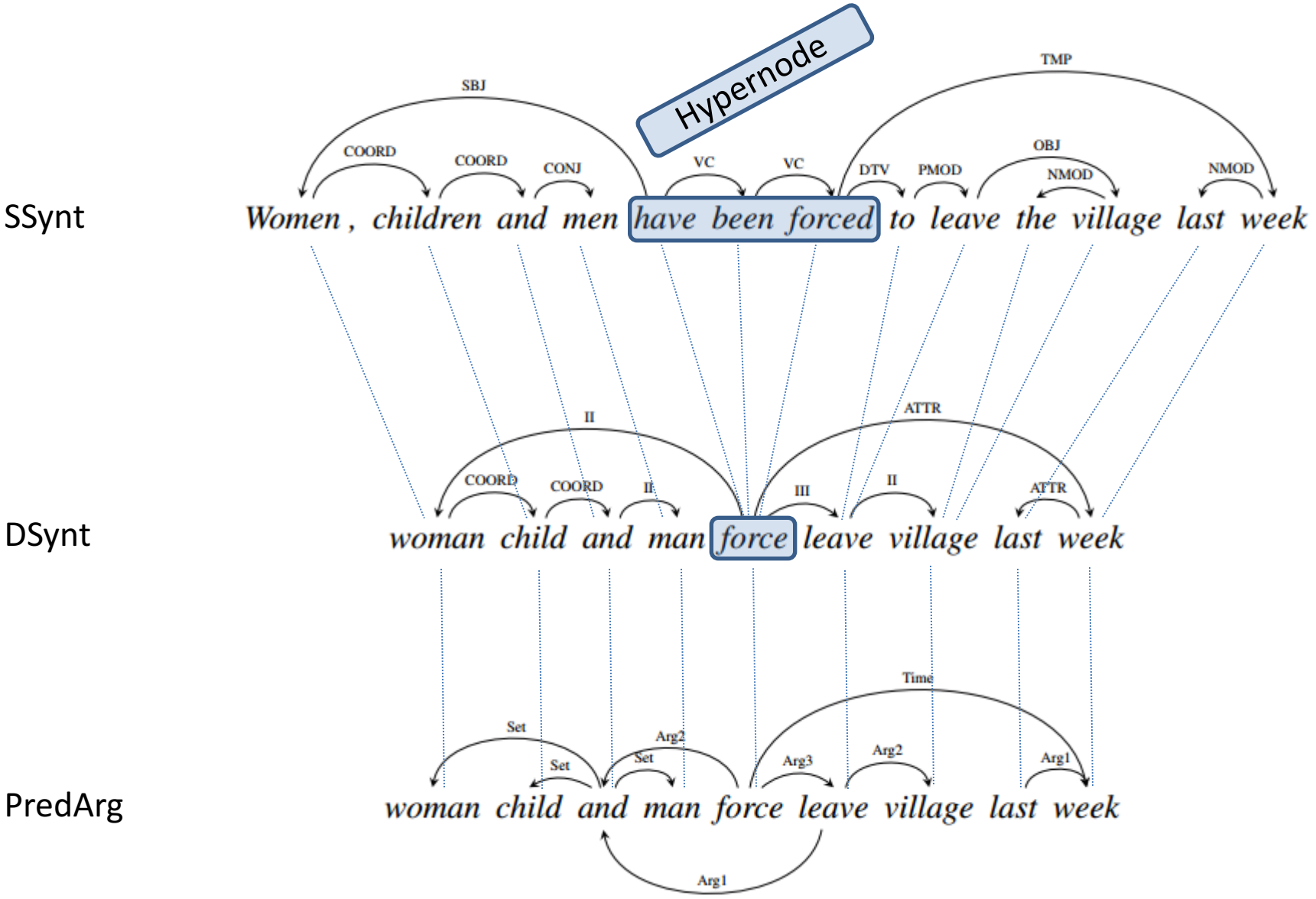
  Trees, fine-grained syntactic DepRels, all words of sentence

- Deep-syntactic output

  Trees, coarse-grained syntactic DepRels, meaning-bearing lexemes

- Predicate-Argument output

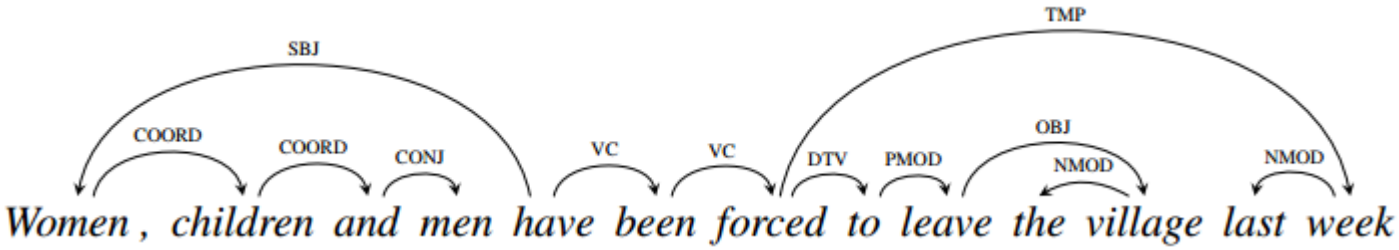
  Graphs, coarse-grained semantico-syntactic DepRels, meaning-bearing lexemes



At each level, we take into consideration **just one type** of phenomenon

but… with several layers, individual annotations are too poor and it makes it all more complicated, doesn't it?

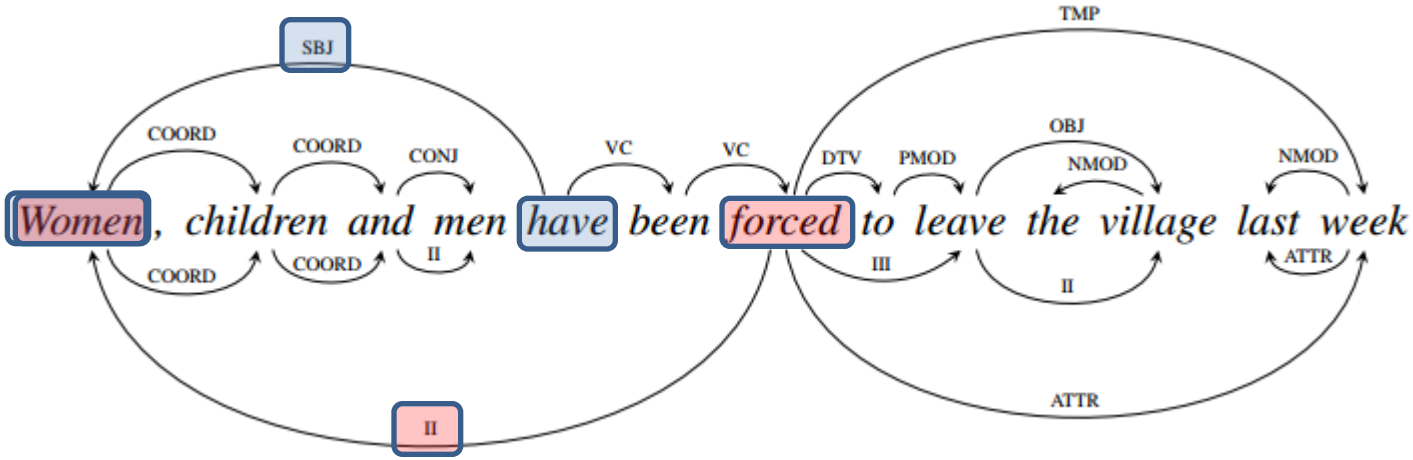
SSynt



boring

some others may say it is clearer this way, and you can always merge the layers together at little cost

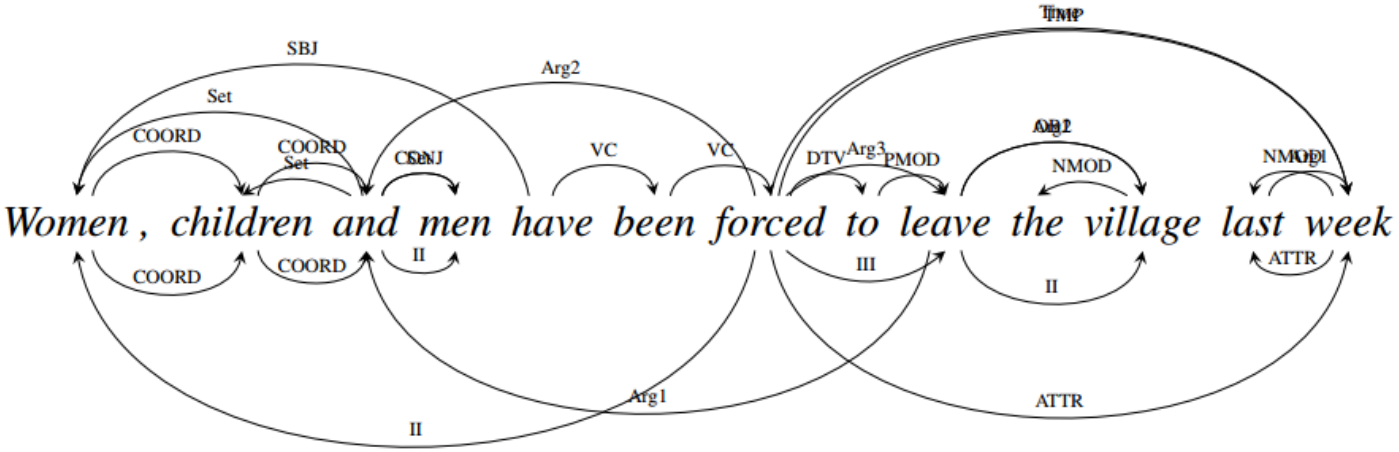
SSynt + DSynt



No constraints of one level onto another level

ah!

SSynt + DSynt + PredArg

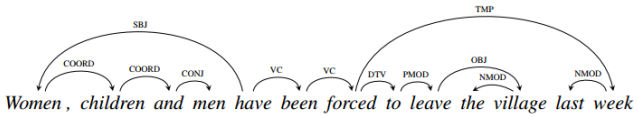


Use what you need!

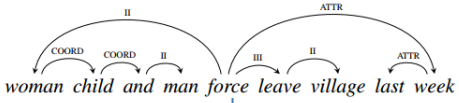
Women, children and men have been forced to leave the village last week.

Module	Toolkit used
Sentence splitting	Stanford Core NLP
Tokenization	Stanford Core NLP
Character normalization	In-house Script
Joint tagging and parsing	(Bohnet and Nivre, 2012)
Speed	≈ 65 ms/sentence
Memory used	≈ 4GB

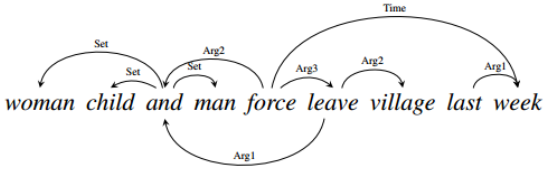
Statistical dependency parsing
(Bohnet and Nivre 2012)
Trained on CoNLL'09 dataset



Rule-based graph-transduction



Rule-based graph-transduction



Grammars	#rul.	Description
ALL	154	
DSynt-Sem	59	Assign core dependencies. Recover shared arguments. Establish coord. conj. as predicates. Assign VerbNet classes.
Post-Proc. 1	11	Recover shared arguments in coordinated constructions. Mark light verbs.
Post-Proc. 2	23	Remove light verbs. Assign frames (FrameNet).
Post-Proc. 3	30	Normalize argument numberings.
Post-Proc. 4	31	Introduce non-core dependencies
Speed	≈ 55 ms/sentence	
Memory used	≈ 300MB	

Grammars	#rul.	Description
ALL	165	
Pre-Proc. 1	15	Assign default PB/NB IDs. Mark passive, genitive, possessive constructions.
Pre-Proc. 2	17	Mark hypernodes.
SSynt-DSynt	55	Wrap hypernodes. Assign DSynt dependencies. Transfer aspect/modality as attr. Mark duplicate relations. Mark relative clauses.
Post-Proc.	78	Relabel duplicate relations. Reestablish gapped elements. Mark coord. constructions.
Speed	≈ 25 ms/sentence	
Memory used	≈ 300MB	

Results

Run	Event	Negation	Opinion	Avg.
SSynt	46.54	59.78	63.62	56.65
DSynt	45.94	33.34	60.42	46.57
PredArg	46.54	30.67	55.86	44.36

- Downstream applications built on syntactic parses
  - SD (Event, Negation) and CoNLL'08 (Opinion)
  - Expected better results for SSynt: number of nodes?
  - Event Extraction seems more insensitive to the type of input
- Similar structures across participants give different results
  - Impact of missing nodes?
  - Impact of number/labels of dependencies?
  - Impact of type of PoS?

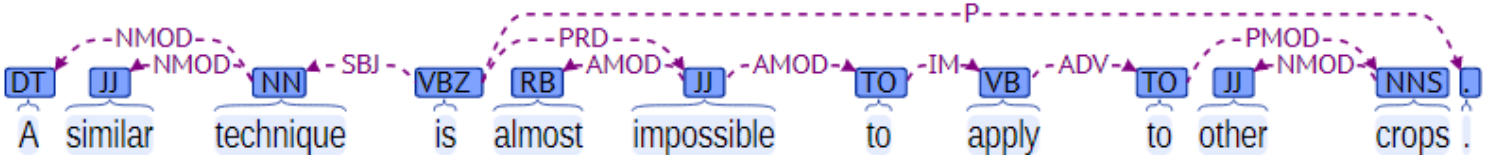
Questions?

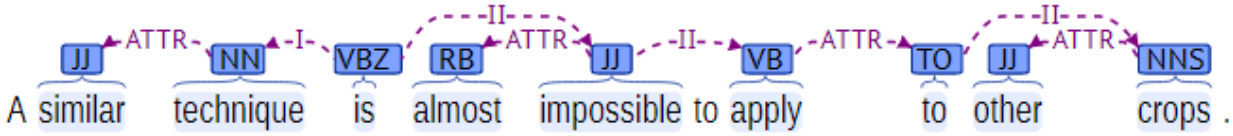
Sample hypernode building rule

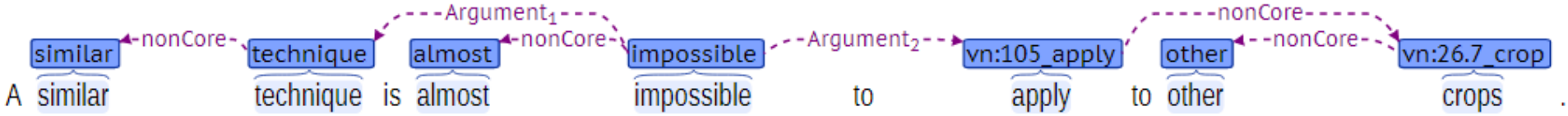
```
c:?XI {
  BLOCK = YES
  c:deprel = ?dep
  c:id = ?i1
  c:?s-> c:?YI {
    c:id = ?i2
  }
}

(?s == PMOD | ?s == IM | ?s == SUB)
```

```
rc:?Yr {
  rc:<=> ?YI
  <=> ?XI
  original_deprel = ?dep
}
```


P
PRD
NMOD
NMOD
SBJ
AMOD
AMOD
IM
ADV
PMOD
NMOD
DT JJ NN VBZ RB JJ TO VB TO JJ NNS .
A similar technique is almost impossible to apply to other crops .


II
ATTR
I
ATTR
II
ATTR
II
ATTR
JJ NN VBZ RB JJ VB TO JJ NNS
A similar technique is almost impossible to apply to other crops .


Argument1
nonCore
nonCore
Argument2
nonCore
nonCore
similar technique almost impossible vn:105_apply other vn:26.7_crop
A similar technique is almost impossible to apply to other crops .