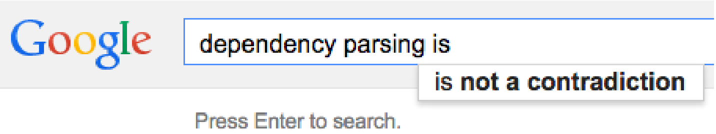
Semantic Dependency Graph Parsing Using Tree Approximations

Željko Agić[♠,♡] Alexander Koller[♡] Stephan Oepen[♣,♡]

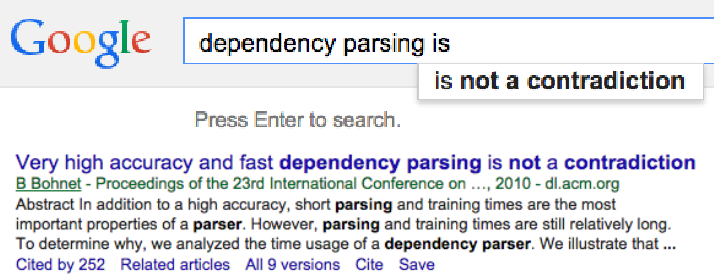
[♠]Center for Language Technology, University of Copenhagen
[♡]Department of Linguistics, University of Potsdam
[♣]Department of Informatics, University of Oslo

IWCS 2015, London, 2015-04-17

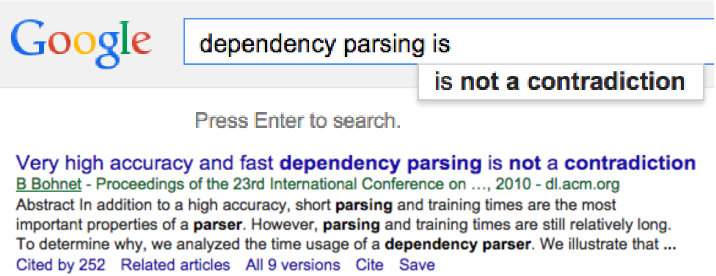
Dependency *tree* parsing



Dependency *tree* parsing

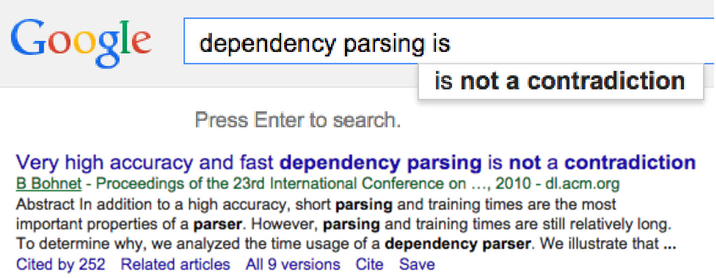


Dependency *tree* parsing



- it is also a big success story in NLP
  - robust and efficient
  - high accuracy across domains and languages
  - enables cross-lingual approaches

Dependency *tree* parsing

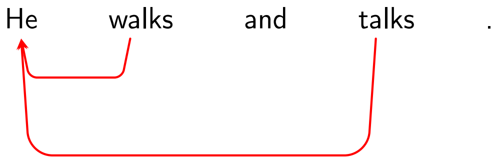


- it is also a big success story in NLP
  - robust and efficient
  - high accuracy across domains and languages
  - enables cross-lingual approaches
- and it is *simple*

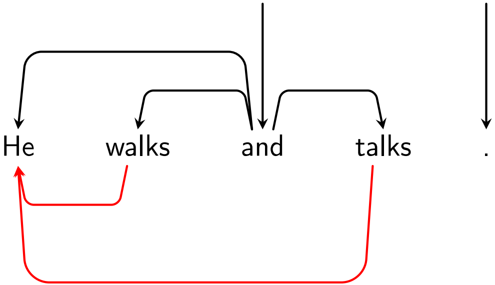
The simplicity

He walks and talks .

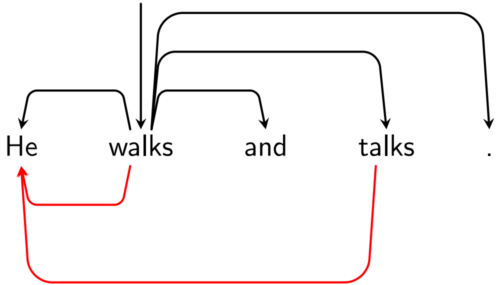
The simplicity



The simplicity



The simplicity



The simplicity

With great speed and accuracy, come great constraints.

- tree constraints
  - single root, single head
  - spanning, connectedness, acyclicity
  - sometimes even projectivity
- there's been a lot of work beyond that
  - plenty of lexical resources
  - successful semantic role labeling shared tasks
  - algorithms for DAG parsing
- but?
  - it's apparently *balkanized*, i.e.,
    the representations are not as uniform as in depparsing

Recent efforts

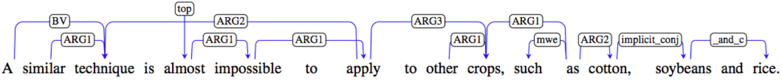
- Banarescu et al. (2013):

  We hope that a sembank of simple, whole-sentence semantic structures will spur new work in statistical natural language understanding and generation, like the Penn Treebank encouraged work on statistical parsing.

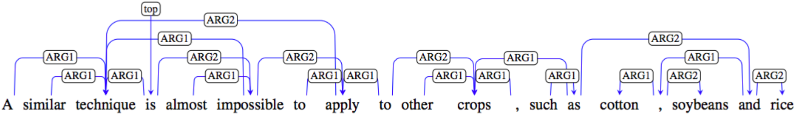
- Oepen et al. (2014):

  SemEval semantic dependency parsing (SDP) shared task
  - WSJ PTB text
  - three DAG annotation layers: DM, PAS, PCEDT
  - bilexical dependencies between words
  - disconnected nodes allowed

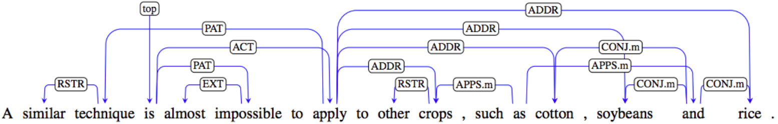
SDP 2014 shared task



(b) DELPH-IN Minimal Recursion Semantics–derived bi-lexical dependencies (DM).



(c) Enju Predicate–Argument Structures (PAS).



(d) Parts of the tectogrammatical layer of the Prague Czech-English Dependency Treebank (PCEDT).

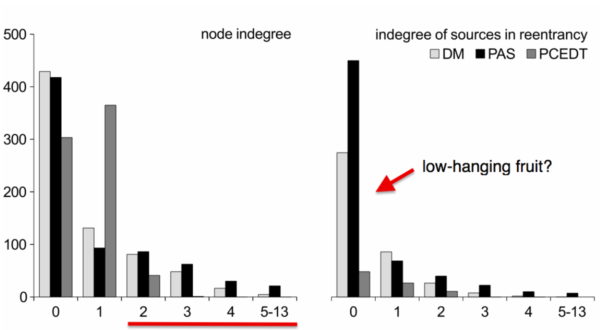
SDP 2014 shared task

		DM	PAS	PCEDT
(1)	**# labels**	51	42	68
(2)	**% singletons**	22.62	4.49	35.79
(3)	**# edge density**	0.96	1.02	0.99
(4)	**$\%_g$ trees**	2.35	1.30	56.58
(5)	**$\%_g$ projective**	3.05	1.71	53.29
(6)	**$\%_g$ fragmented**	6.71	0.23	0.56
(7)	**$\%_n$ reentrancies**	27.35	29.40	9.27
(8)	**$\%_g$ topless**	0.28	0.02	0.00
(9)	**# top nodes**	0.9972	0.9998	1.1237
(10)	**$\%_n$ non-top roots**	44.71	55.92	4.36

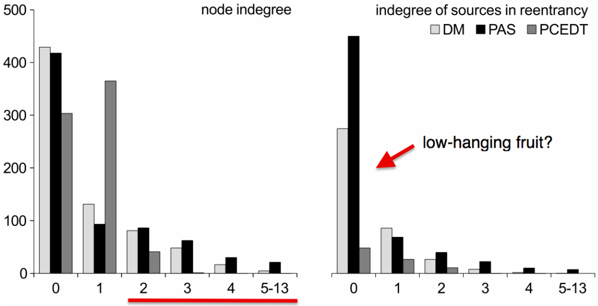
	Directed			**Undirected**		
	DM	PAS	PCEDT	DM	PAS	PCEDT
DM	–	.6425	.2612	–	.6719	.5675
PAS	.6688	–	.2963	.6993	–	.5490
PCEDT	.2636	.2963	–	.5743	.5630	–

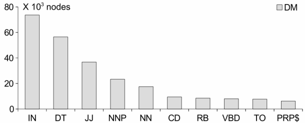
- uniform, but not the same
- PCEDT seems to be somewhat more distinct
- key ingredients of non-trees
  - singletons
  - reentrancies: *indegree* > 1

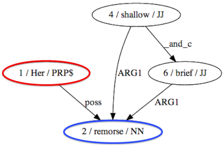
Reentrancies



Reentrancies





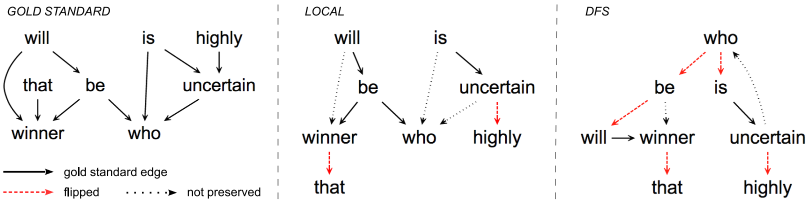


Parsing with tree approximations

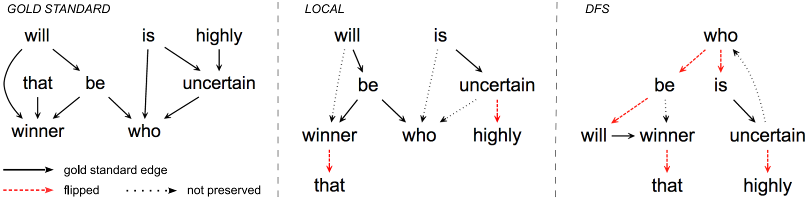
Team	Track	Approach	Resources
Linköping	C	extension of Eisner's algorithm for DAGs, edge-factored structured perceptron	—
Potsdam	C & O	graph-to-tree transformation, Mate	companion
Priberam	C & O	model with second-order features, decoding with dual decomposition, MIRA	companion
Turku	O	cascade of SVM classifiers (dependency recognition, label classification, top recognition)	companion, syntactic n-grams, word2vec
Alpage	C & O	transition-based parsing for DAGs, logistic regression, structured perceptron	companion, Brown clusters
Peking	C	transition-based parsing for DAGs, graph-to-tree transformation, parser ensemble	—
CMU	O	edge classification by logistic regression, edge-factored structured SVM	companion
Copenhagen-Malmö	C	graph-to-tree transformation, Mate	—
In-House	O	existing parsers developed by the organizers	grammars

Hey, these DAGs are very tree-like.
Let's convert them to trees and use standard depparsers!

Parsing with tree approximations

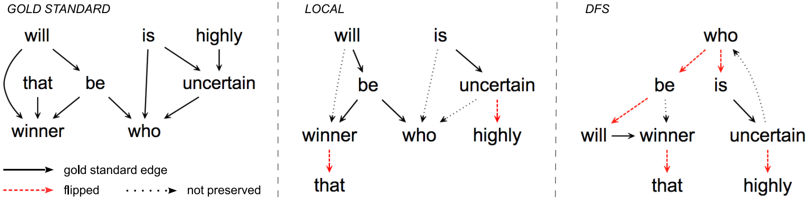


Parsing with tree approximations



- flip the flippable, baseline-delete the rest
- train on trees, parse for trees, flip back in post-processing

Parsing with tree approximations



- flip the flippable, baseline-delete the rest
- train on trees, parse for trees, flip back in post-processing

- works OK...ish
  - average labeled F_1 in the high 70s
  - task winner votes between tree approximations

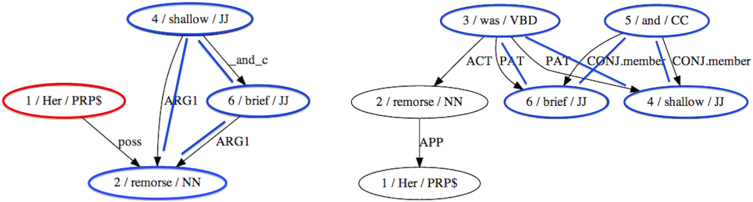
Where do all the lost edges go?

- the deleted edges cannot be recovered
- upper bound recall
  - graph-tree-graph conversion with no parsing in-between
  - measure the lossiness

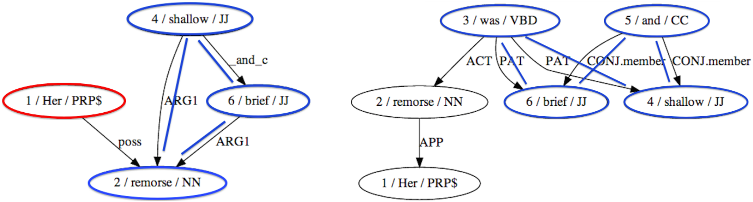
	DM				PCEDT			
	P	*R*	*LM*	# labels	*P*	*R*	*LM*	# labels
OFFICIAL	100.00	55.28	2.54	52	100.00	90.35	54.33	71
LOCAL	100.00	87.50	17.35	79	100.00	92.33	54.65	124
DFS	100.00	97.30	65.43	79	100.00	94.03	54.58	133

- new agenda
  - inspect the lost edges
  - build a better tree approximation on top

Where do all the lost edges go?

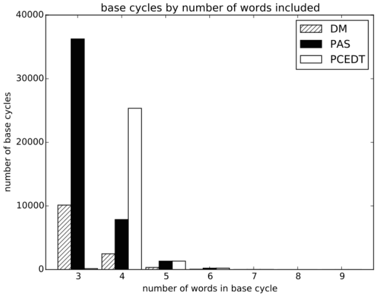


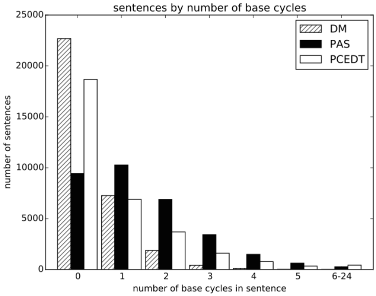
Where do all the lost edges go?



- there are *undirected cycles* in the graphs
  - interesting structural properties?
  - discriminate specific phenomena they encode?

Undirected cycles





- we mostly ignore PAS from now on
- DM: 3-word cycles dominate *(triangles)*
- PCEDT: 4-word cycles *(squares)*
- sentences with more than one cycle not very frequent

Undirected cycles

DM			PAS			PCEDT		
	#	%		#	%		#	%
N V V	3843	29.63	N V V	15541	34.44	CC N N V	4789	17.72
PRP V V	1208	9.31	MD N V	5005	11.09	CC N N N	3418	12.65
N TO V V	1203	9.28	PRP V V	4012	8.89	, N N V	2512	9.29
J N V	1059	8.16	J N V	3544	7.85	CC V V V	1633	6.04
IN N V	962	7.42	CC N V V	2155	4.78	CC N V V	1614	5.97
J J N	506	3.90	MD PRP V	1622	3.59	N N N V	805	2.98
CD CD N	324	2.50	IN N V	1087	2.41	N N V V	752	2.78
PRP TO V V	277	2.14	J PRP V	877	1.94	N V V V	665	2.46
J PRP V	228	1.76	CC N N N	676	1.50	, N N N	495	1.83
N N V	202	1.56	CC V V	561	1.24	CC J J N	447	1.65

- DM, PAS: mostly control and coordination
- PCEDT: almost exclusively coordination
- supported also by the edge label tuples, and the lemmas

Back to tree approximations

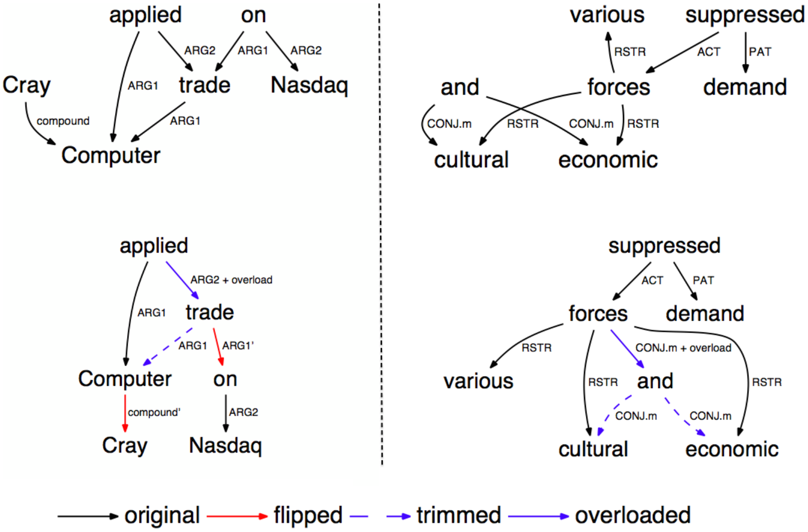
- edge operations up to now
  - *flipping* – comes with implicit *overloading*
  - *deletion* – edges are permanently lost

Back to tree approximations

- edge operations up to now
  - *flipping* – comes with implicit *overloading*
  - *deletion* – edges are permanently lost
- new proposal
  - detect an undirected cycle
  - select and disconnect an appropriate edge
    - *radical:* overload an appropriate label for reconstruction, or
    - *conservative:* trim only a subset of edges using lemma-POS cues
  - in post-processing, reconnect the edge
    - by reading the reconstruction off of the overloaded label, or
    - by detecting the lemma-POS trigger

  - we call these operations *trimming* and *untrimming*

Trimming and untrimming



Upper bounds

	DM				PCEDT			
	P	*R*	*LM*	# labels	*P*	*R*	*LM*	# labels
OFFICIAL	100.00	55.28	2.54	52	100.00	90.35	54.33	71
LOCAL	100.00	87.50	17.35	79	100.00	92.33	54.65	124
DFS	100.00	97.30	65.43	79	100.00	94.03	54.58	133
radical trimming								
▽ + LOCAL	100.00	88.33	21.07	101	–	–	–	–
▽ + DFS	100.00	98.89	85.07	154	–	–	–	–
□ + LOCAL	–	–	–	–	100.00	93.59	56.02	382
□ + DFS	–	–	–	–	100.00	95.21	66.33	413
conservative trimming								
▽ + LOCAL	98.98	87.93	19.66	79	–	–	–	–
▽ + DFS	99.12	98.07	83.83	79	–	–	–	–
□ + LOCAL	–	–	–	–	98.83	92.88	54.99	124
□ + DFS	–	–	–	–	98.96	94.65	65.57	133
radical – DFS	0.00	+1.59	+19.64	+75	0.00	+1.18	+11.75	+280
conservative – DFS	-0.88	+0.77	+18.40	0	-1.04	+0.62	+10.99	0

Parsing

- preprocessing: trimming + DFS + baseline = training trees
- training and parsing
  - `mate-tools` graph-based depparser
  - CRF++ for top node detection
  - SDP companion data and Brown clusters as additional features
- postprocessing: removing baseline artifacts + reflipping + + untrimming = output graphs

Results

	closed track						*open track*					
	DM			PCEDT			DM			PCEDT		
	LF	*LM*	*LAS*	*LF*	*LM*	*LAS*	*LF*	*LM*	*LAS*	*LF*	*LM*	*LAS*
DFS	79.35	9.05	78.99	67.92	5.86	81.01	83.00	10.46	84.00	70.24	5.79	85.44
radical												
▽ + DFS	77.73	12.15	75.62	–	–	–	80.56	13.44	80.23	–	–	–
□ + DFS	–	–	–	65.33	6.67	77.47	–	–	–	66.14	6.98	83.37
conservative												
▽ + DFS	80.05	18.91	79.04	–	–	–	83.55	20.01	83.96	–	–	–
□ + DFS	–	–	–	68.82	11.53	81.05	–	–	–	71.18	12.09	85.53
radical – DFS	-1.62	3.10	-3.37	-2.59	0.81	-3.54	-2.44	2.98	-3.77	-4.10	1.19	-2.07
conservative – DFS	0.70	9.86	+0.05	0.90	5.67	+0.04	0.55	9.55	-0.04	0.94	6.30	0.09

- lower upper bounds, higher parsing scores
- nice increase in *LM*
- best overall score for any tree approximation-based system

Conclusions

- our contributions
  - put SDP DAGs under the lens
  - uncovered the link between non-trees and control, coordination
  - used this to implement a
    state-of-the-art system based on tree approximations
- future work
  - did some more experiments
    - answer set programming for better tree approximations
    - did not see improvements
  - go for *real* graph parsing

Thank you for your attention. ☺