Arc-Standard Spinal Parsing with Stack-LSTMs

Miguel Ballesteros

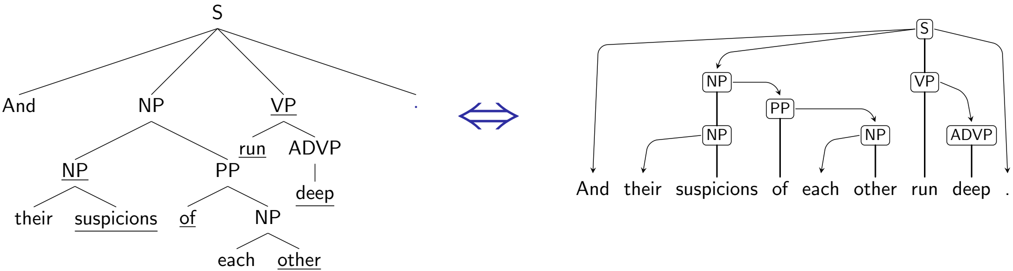
IBM Research

Xavier Carreras

NAVER LABS

IWPT 2017

Spinal Trees = Constituency Trees + Dependency Heads

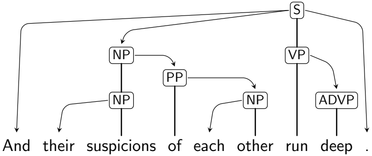


See: Carreras, Collins, and Koo (2008)

See also: Collins (1997); Ballesteros and Carreras (2015); Hayashi, Suzuki, and Nagata (2016)

In this paper . . .

- Arc-standard system for Spinal Parsing
- Stack-LSTMs Dyer et al. (2015)



Outline

Introduction

Arc-Standard Spinal Parsing

Stack LSTMs for Spinal Parsing

Experiments

Conclusions

Outline

Introduction

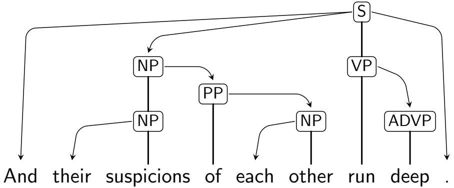
Arc-Standard Spinal Parsing

Stack LSTMs for Spinal Parsing

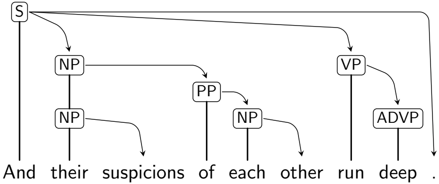
Experiments

Conclusions

Spinal Trees



(stanford dependencies)



(leftmost heads)

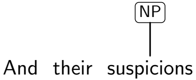
- Spine: a vertical sequence of non-terminals
- Spinal Trees: spines connected by dependencies
- Spinal tree ⇔ Constituent tree + Dependency tree

Arc-Standard Spinal Parsing

Extending Nivre (2004) . . .

Actions $\begin{cases} \texttt{shift token} \\ \texttt{add spinal node} \text{ (new for spinal parsing)} \\ \texttt{left-arc} \text{ and } \texttt{right-arc} \end{cases}$

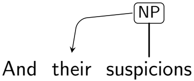
Transition	Buffer β	Stack σ	New Arc in δ
	[And, their, . . .]	[]	
shift	[their, suspicions, . . .]	[And]	
shift	[suspicions, of, . . .]	[And, their]	
shift	[of, each, . . .]	[. . . , their, susp.]	
node(NP)	[of, each, . . .]	[. . . , their, susp.+NP_3^1]	



Arc-Standard Spinal Parsing

Extending Nivre (2004) ...

Actions
- `shift token`
- `add spinal node` (new for spinal parsing)
- `left-arc` and `right-arc`

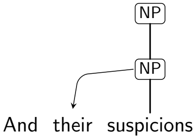


Transition	Buffer β	Stack σ	New Arc in δ
	[And, their, ...]	[]	
`shift`	[their, suspicions, ...]	[And]	
`shift`	[suspicions, of, ...]	[And, their]	
`shift`	[of, each, ...]	[..., their, susp.]	
`node(NP)`	[of, each, ...]	[..., their, susp.+NP^1_3]	
`left-arc`	[of, each, ...]	[And, susp.+NP^1_3]	(NP^1_3,their)

Arc-Standard Spinal Parsing

Extending Nivre (2004) . . .

Actions {
- `shift token`
- `add spinal node` (new for spinal parsing)
- `left-arc` and `right-arc`

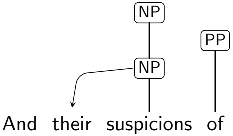


Transition	Buffer β	Stack σ	New Arc in δ
	[And, their, . . .]	[]	
`shift`	[their, suspicions, . . .]	[And]	
`shift`	[suspicions, of, . . .]	[And, their]	
`shift`	[of, each, . . .]	[. . . , their, susp.]	
`node(NP)`	[of, each, . . .]	[. . . , their, susp.+NP$^1_3$]	
`left-arc`	[of, each, . . .]	[And, susp.+NP$^1_3$]	(NP$^1_3$,their)
`node(NP)`	[of, each, . . .]	[And, susp.+NP$^1_3$+NP$^2_3$]	

Arc-Standard Spinal Parsing

Extending Nivre (2004) ...

Actions {
- `shift token`
- `add spinal node` (new for spinal parsing)
- `left-arc` and `right-arc`

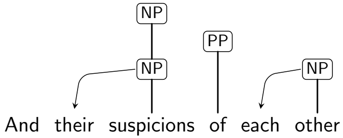


Transition	Buffer β	Stack σ	New Arc in δ
	[And, their, ...]	[]	
`shift`	[their, suspicions, ...]	[And]	
`shift`	[suspicions, of, ...]	[And, their]	
`shift`	[of, each, ...]	[..., their, susp.]	
`node(NP)`	[of, each, ...]	[..., their, susp.+NP^1_3]	
`left-arc`	[of, each, ...]	[And, susp.+NP^1_3]	(NP^1_3,their)
`node(NP)`	[of, each, ...]	[And, susp.+NP^1_3+NP^2_3]	
`shift`	[each, other, ...]	[..., susp.+NP^1_3+NP^2_3, of]	
`node(PP)`	[each, other, ...]	[..., susp.+NP^1_3+NP^2_3, of+PP^1_4]	

Arc-Standard Spinal Parsing

Extending Nivre (2004) . . .

Actions
- `shift token`
- `add spinal node` (new for spinal parsing)
- `left-arc` and `right-arc`

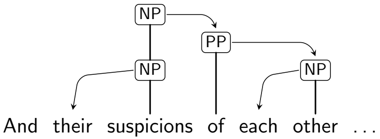


Transition	Buffer β	Stack σ	New Arc in δ
	[And, their, . . .]	[]	
`shift`	[their, suspicions, . . .]	[And]	
`shift`	[suspicions, of, . . .]	[And, their]	
`shift`	[of, each, . . .]	[. . . , their, susp.]	
`node(NP)`	[of, each, . . .]	[. . . , their, susp.+NP$_3^1$]	
`left-arc`	[of, each, . . .]	[And, susp.+NP$_3^1$]	(NP$_3^1$,their)
`node(NP)`	[of, each, . . .]	[And, susp.+NP$_3^1$+NP$_3^2$]	
`shift`	[each, other, . . .]	[. . . , susp.+NP$_3^1$+NP$_3^2$, of]	
`node(PP)`	[each, other, . . .]	[. . . , susp.+NP$_3^1$+NP$_3^2$, of+PP$_4^1$]	
`shift`	[other, run, . . .]	[. . . , of+PP$_4^1$, each]	
`shift`	[run, deep, . . .]	[. . . , each, other]	
`node(NP)`	[run, deep, . . .]	[. . . , each, other+NP$_6^1$]	
`left-arc`	[run, deep, . . .]	[. . . , of+PP$_4^1$, other+NP$_6^1$]	(NP$_6^1$, each)

Arc-Standard Spinal Parsing

Extending Nivre (2004) . . .

Actions
- `shift token`
- `add spinal node` (new for spinal parsing)
- `left-arc` and `right-arc`



Transition	Buffer β	Stack σ	New Arc in δ
	[And, their, . . .]	[]	
`shift`	[their, suspicions, . . .]	[And]	
`shift`	[suspicions, of, . . .]	[And, their]	
`shift`	[of, each, . . .]	[. . . , their, susp.]	
`node(NP)`	[of, each, . . .]	[. . . , their, susp.+NP$^{1}_{3}$]	
`left-arc`	[of, each, . . .]	[And, susp.+NP$^{1}_{3}$]	(NP$^{1}_{3}$,their)
`node(NP)`	[of, each, . . .]	[And, susp.+NP$^{1}_{3}$+NP$^{2}_{3}$]	
`shift`	[each, other, . . .]	[. . . , susp.+NP$^{1}_{3}$+NP$^{2}_{3}$, of]	
`node(PP)`	[each, other, . . .]	[. . . , susp.+NP$^{1}_{3}$+NP$^{2}_{3}$, of+PP$^{1}_{4}$]	
`shift`	[other, run, . . .]	[. . . , of+PP$^{1}_{4}$, each]	
`shift`	[run, deep, . . .]	[. . . , each, other]	
`node(NP)`	[run, deep, . . .]	[. . . , each, other+NP$^{1}_{6}$]	
`left-arc`	[run, deep, . . .]	[. . . , of+PP$^{1}_{4}$, other+NP$^{1}_{6}$]	(NP$^{1}_{6}$, each)
`right-arc`	[run, deep, . . .]	[. . . , susp.+NP$^{1}_{3}$+NP$^{2}_{3}$, of+PP$^{1}_{4}$]	(PP$^{1}_{4}$, NP$^{1}_{6}$)
`right-arc`	[run, deep, . . .]	[And, susp.+NP$^{1}_{3}$+NP$^{2}_{3}$]	(NP$^{2}_{3}$, PP$^{1}_{4}$)
. . .			

Arc-Standard Spinal Parsing: Properties

- Like arc-standard dependency parsing Nivre (2008), it builds partial trees bottom-up
    - Thus, it is straightforward to add spinal nodes
- Same transition system proposed by Cross and Huang (2016)
    - They turn shift-reduce constituent parsing to be head-driven (i.e. spinal)
- Previous work on transition-based spinal parsing predicted full spines all at once
    - Long-tail distribution
    - See Ballesteros and Carreras (2015); Hayashi et al. (2016)

Outline

Introduction

Arc-Standard Spinal Parsing

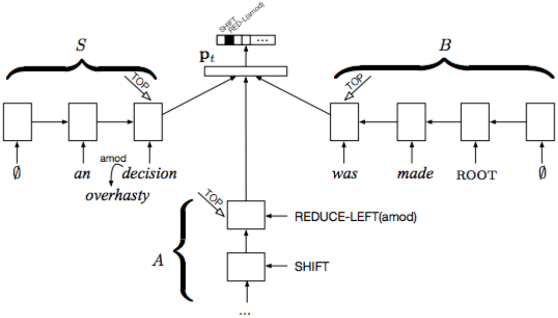
Stack LSTMs for Spinal Parsing

Experiments

Conclusions

Stack LSTMs

by Dyer, Ballesteros, Ling, Matthews, and Smith (2015)



(figure from Dyer et al. (2015))

Stack LSTMs compute the parser's state combining LSTMs on three stacks: parser stack (S), buffer (B), and action list (A).

Stack LSTMs for Spinal Parsing

Stack LSTMs maintain a state vector $\mathbf{c} \in \mathbb{R}^k$ (i.e. embedding) for each element in the stack. We extend it to spinal parsing.

shift(t)

$$\mathbf{c} = \mathbf{w}_t$$

where:

- t is a token
- $\mathbf{w}_t$ is an embedding (combines word and pos-tag)

left/right arc(h, d)

$$\mathbf{c} = \tanh(\mathbf{U}[\mathbf{h}; \mathbf{d}] + \mathbf{e})$$

where:

- $\mathbf{h}$ is the embedding of the head tree
- $\mathbf{d}$ is the embedding of the dependent tree
- $\mathbf{U}$ and $\mathbf{e}$ are parameter weights and bias terms

node(s, n)

$$\mathbf{c} = \tanh(\mathbf{W}[\mathbf{s}; \mathbf{n}] + \mathbf{b})$$

where:

- $\mathbf{s}$ is the embedding of the current spine
- $\mathbf{n}$ is the embedding of the new node
- $\mathbf{U}$ and $\mathbf{b}$ are parameter weights and bias terms

Outline

Introduction

Arc-Standard Spinal Parsing

Stack LSTMs for Spinal Parsing

Experiments

Conclusions

Experiments

Questions:

1. In constituency parsing, what head-rules are more helpful?
2. In dependency parsing, is constituent structure helpful?

Setting:

- Penn Treebank parsing, standard splits
- Word embeddings from (Dyer et al., 2015)
- Head rules:
  - Stanford Dependencies by De Marneffe, MacCartney, Manning, et al. (2006)
  - Yamada and Matsumoto (2003)
  - Leftmost heads, Rightmost heads

Experiments on Development Data

	LR	LP	F1	UAS
Leftmost heads	91.18	90.93	91.05	-
Rightmost heads	91.03	91.20	91.11	-
SD heads	90.75	91.11	90.93	93.49
YM heads	90.82	90.84	90.83	-

- Rightmost and Leftmost heads improve over linguistically-motivated head rules

Experiments on Development Data

	LR	LP	F1	UAS
Leftmost heads	91.18	90.93	91.05	-
↪ no node comp.	90.20	90.76	90.48	-
Rightmost heads	91.03	91.20	91.11	-
↪ no node comp.	90.64	91.24	90.04	-
SD heads	90.75	91.11	90.93	93.49
↪ no node comp.	90.38	90.58	90.48	93.16
YM heads	90.82	90.84	90.83	-

- Rightmost and Leftmost heads improve over linguistically-motivated head rules
- Node composition in the Stack LSTMs is helpful

Experiments on Development Data

	LR	LP	F1	UAS
Leftmost heads	91.18	90.93	91.05	-
↪ no node comp.	90.20	90.76	90.48	-
Rightmost heads	91.03	91.20	91.11	-
↪ no node comp.	90.64	91.24	90.04	-
SD heads	90.75	91.11	90.93	93.49
↪ no node comp.	90.38	90.58	90.48	93.16
↪ dummy spines	-	-	-	93.30
YM heads	90.82	90.84	90.83	-

- Rightmost and Leftmost heads improve over linguistically-motivated head rules
- Node composition in the Stack LSTMs is helpful
- Consituent structure is helpful

Experiments: Constituency Results on PTB Test

	LR	LP	F1
Spinal (leftmost)	90.30	90.54	90.42
Spinal (rightmost)	90.23	90.77	90.50
Ballesteros and Carreras (2015)	88.7	89.2	89.0
Carreras et al. (2008)	90.7	91.4	91.1
Vinyals et al. (2014) (PTB-Only)			88.3
Cross and Huang (2016)			89.9
Choe and Charniak (2016) (PTB-Only)			91.2
Choe and Charniak (2016) (Semi-sup)			93.8
Dyer et al. (2016) (Discr.)			91.2
Dyer et al. (2016) (Gen.)			93.3
Kuncoro et al. (2017) (Gen.)			93.5
Liu and Zhang (2017a)	91.3	92.1	91.7
Liu and Zhang (2017b)			90.5
Liu and Zhang (2017b) (semi-rerank)			93.4

Experiments: Dependency Results on PTB Test

	UAS test
Spinal, PTB spines + SD (TB-greedy)	93.15
Spinal, dummy spines + SD (TB-greedy)	93.10
Dyer et al. (2015) (TB-greedy)	93.1
Cross and Huang (2016)	93.4
Ballesteros et al. (2016) (TB-dynamic)	93.6
Kiperwasser and Goldberg (2016) (TB-dynamic)	93.9
Andor et al. (2016) (TB-Beam)	94.6
Kuncoro et al. (2016) (Graph-Ensemble)	94.5
Choe and Charniak (2016)* (Semi-sup)	95.9
Kuncoro et al. (2017)* (Generative)	95.8
Dozat and Manning (2016)*	95.7

Outline

Introduction

Arc-Standard Spinal Parsing

Stack LSTMs for Spinal Parsing

Experiments

Conclusions

Conclusions

- Contribution: Spinal Stack LSTMs for Dependency+Constituent Parsing
  - Simple extension to the arc-standard system (i.e add node)
  - In general, it is handy to model both constituents and dependencies
  - In practice, Stack-LSTMs do not seem to benefit from linguistic head rules
- Simple extensions:
  - UD
  - Adding dependency labels
  - Other languages
- More interestingly:
  - What kind of hidden structure do LSTMs capture?
  - Is it similar, better, or complementary to that given by head rules?

References I

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2442–2452, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1231.

Miguel Ballesteros and Xavier Carreras. Transition-based spinal parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 289–299. Association for Computational Linguistics, 2015. doi: 10.18653/v1/K15-1029. URL http://aclweb.org/anthology/K15-1029.

Miguel Ballesteros, Yoav Goldberg, Chris Dyer, and Noah A. Smith. Training with exploration improves a greedy stack lstm parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2005–2010, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1211.

Xavier Carreras, Michael Collins, and Terry Koo. *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, chapter TAG, Dynamic Programming, and the Perceptron for Efficient, Feature-Rich Parsing, pages 9–16. Coling 2008 Organizing Committee, 2008. URL http://aclweb.org/anthology/W08-2102.

Do Kook Choe and Eugene Charniak. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2331–2336, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1257.

Michael Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain, July 1997. Association for Computational Linguistics. doi: 10.3115/976909.979620. URL http://www.aclweb.org/anthology/P97-1003.

James Cross and Liang Huang. Incremental parsing with minimal features using bi-directional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 32–37. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-2006. URL http://aclweb.org/anthology/P16-2006.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454. Genoa, 2006.

Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. *CoRR*, abs/1611.01734, 2016. URL http://arxiv.org/abs/1611.01734.

References II

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China, July 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P15-1033.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California, June 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N16-1024.

Katsuhiko Hayashi, Jun Suzuki, and Masaaki Nagata. Shift-reduce spinal tag parsing with dynamic programming. *Transactions of the Japanese Society for Artificial Intelligence*, 31(2), 2016.

Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327, 2016. ISSN 2307-387X. URL https://transacl.org/ojs/index.php/tacl/article/view/885.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, and Noah A. Smith. Distilling an ensemble of greedy dependency parsers into one mst parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1744–1753, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1180.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A. Smith. What do recurrent neural network grammars learn about syntax? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1249–1258, Valencia, Spain, April 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E17-1117.

Jiangming Liu and Yue Zhang. Shift-reduce constituent parsing with neural lookahead features. *Transactions of the Association of Computational Linguistics*, 5:45–58, 2017a. URL http://aclanthology.coli.uni-saarland.de/pdf/Q/Q17/Q17-1004.pdf.

Jiangming Liu and Yue Zhang. Encoder-decoder shift-reduce syntactic parsing. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 105–114, Pisa, Italy, September 2017b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W17-6315.

References III

Joakim Nivre. Incrementality in deterministic dependency parsing. In Frank Keller, Stephen Clark, Matthew Crocker, and Mark Steedman, editors, *Proceedings of the ACL Workshop Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50–57, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Joakim Nivre. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553, December 2008. ISSN 0891-2017.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey E. Hinton. Grammar as a foreign language. *CoRR*, abs/1412.7449, 2014. URL http://arxiv.org/abs/1412.7449.

Hiroyasu Yamada and Yuji Matsumoto. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*, volume 3, pages 195–206, 2003.