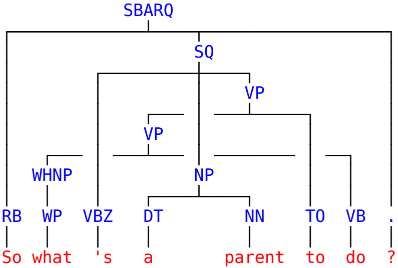
Discontinuous Constituency Parsing with a Stack-Free Transition System and a Dynamic Oracle

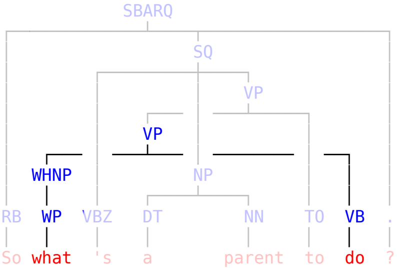
Maximin Coavoux[1] – Shay B. Cohen[2]

[1]Naver Labs Europe – [2]University of Edinburgh

NAACL 2019

Task: Discontinuous constituency parsing

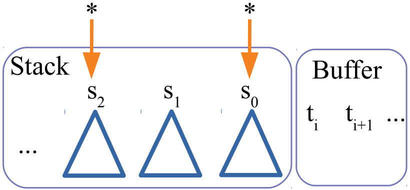




- Discontinuous constituents for representing, e.g.:
  - Long distance extractions: relative clauses, questions
  - Cross serial dependencies
- → phenomena hard to represent with projective constituents (and usually ignored by projective constituency parsers)

Prior work – Transition-Based Parsing

- Shift-reduce paradigm: parsers store subtrees in a **stack**
- Discontinuous parsing: need to access older elements in the stack to construct discontinuous constituents.
  - Dedicated actions: SWAP (Maier, 2015), GAP (Coavoux and Crabbé, 2017)
  - Stack = linear time access: **need *n* operations to access *n*th element**

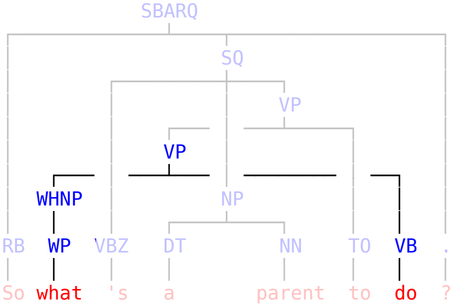


Proposal

- Stack-free transition system: the parser uses an **unordered memory** to store subtrees instead of a stack: **constant time access** to any element in the stack
- Dynamic oracle for the transition system (improve training of parser)

Stack-Free Transition System: Tree Representation

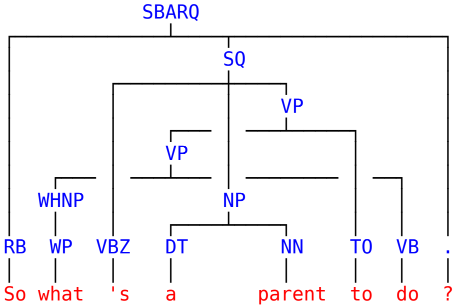
- Discontinuous tree = set of constituents
- Constituents = labelled set of tokens



$$\equiv \left\{ \begin{array}{l} (\text{SBARQ}, \{\text{So, what, 's, a, parent, to, do, ?}\}) \\ (\text{SQ}, \{\text{what, 's, a, parent, to, do}\}) \\ (\text{VP}, \{\text{what, do}\}) \\ (\text{VP}, \{\text{what, to, do}\}) \\ (\text{NP}, \{\text{a, parent}\}) \\ (\text{WHNP}, \{\text{what}\}) \end{array} \right\}$$

Stack-Free Transition System: Tree Representation

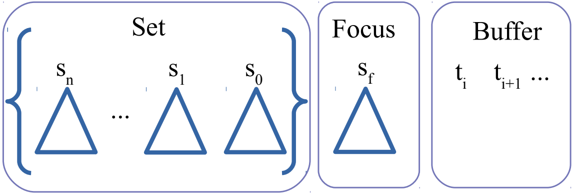
- Discontinuous tree = set of constituents
- Constituents = labelled set of tokens



$$\equiv \left\{ \begin{array}{l} (\text{SBARQ}, \{\text{So, what, 's, a, parent, to, do, ?}\}) \\ (\text{SQ}, \{\text{what, 's, a, parent, to, do}\}) \\ (\text{VP}, \{\text{what, do}\}) \\ (\text{VP}, \{\text{what, to, do}\}) \\ (\text{NP}, \{\text{a, parent}\}) \\ (\text{WHNP}, \{\text{what}\}) \end{array} \right\}$$

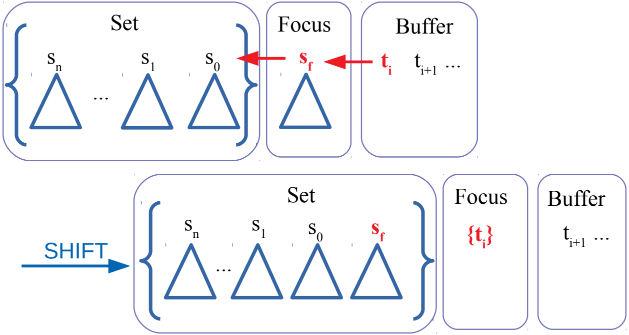
- unified representation for both projective and discontinuous constituents

Stack-Free Transition System: Configuration



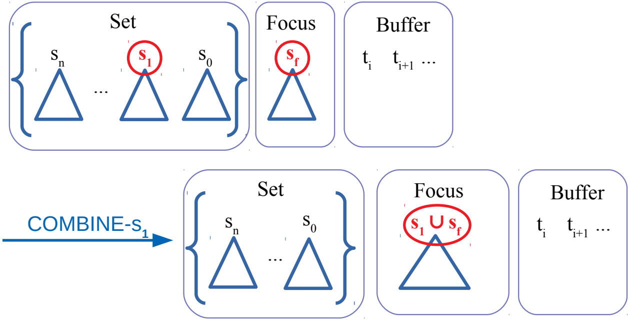
- Subtrees are stored in a **set** of parsing items
  - Each parsing item s_i is a set of tokens
- The **focus item** is a distinguished item s_f
  - Invariant: the focus item always contains **the last token that has been shifted**
- Remaining tokens are stored in a **buffer**
- Actions: **Shift**, **Combine-*s***, **Label-X**, **Nolabel**

Stack-Free Transition System: Actions – Shift



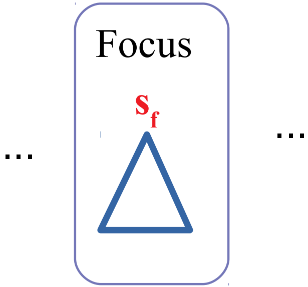
- Current focus item is added to the memory
- New focus item is the shifted token

Stack-Free Transition System: Actions – Combine-*s*



- Combine-*s* is parameterized by an item *s* in the memory
  - *n* items in the memory ⇒ *n* potential Combine
- Bottom-up combination: compute the union between the focus item s_f and s_i
- The result of the union becomes the **new focus item**

Stack-Free Transition System: Actions – Label-X / Nolabel



- label-X: instantiates a constituent (X, s_f)
  - X is a non terminal
- nolabel: do nothing

Stack-Free Transition System: Properties

- Alternation between **structural** (shift, combine) and **labelling** (label-X, nolabel) actions (Cross and Huang, 2016)

  $$((\text{shift} \mid \text{combine})(\text{label-X} \mid \text{nolabel}))^{2n-1}$$

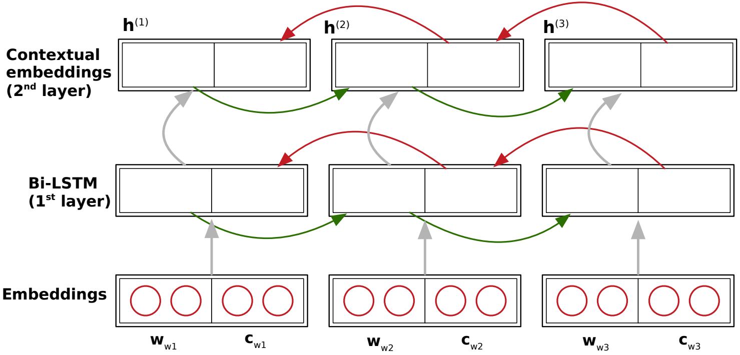
  → simpler atomic decisions

- Derives any labelled discontinuous tree in exactly $4n-2$ actions:
  - n shifts
  - $n-1$ combine
  - $2n-1$ labelling actions (1 after each structural action)

- Supports a **dynamic oracle**, building upon work by Cross and Huang (2016) for projective parsing (see paper for details)
  - Dynamic oracle: function required to train the parser on any potential configurations as opposed to gold configuration only (static oracle = teacher forcing)

Scoring System: Token Representations

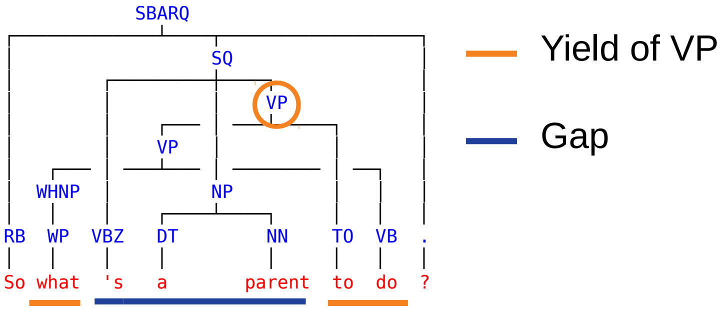
Token vectors: contextual embeddings (2-layer bi-LSTM)



- $\mathbf{w}_w$: word embedding
- $\mathbf{c}_w = \text{bi-LSTM}(w)$: char-based embedding (character bi-LSTM)

Scoring System: Constituent Representations

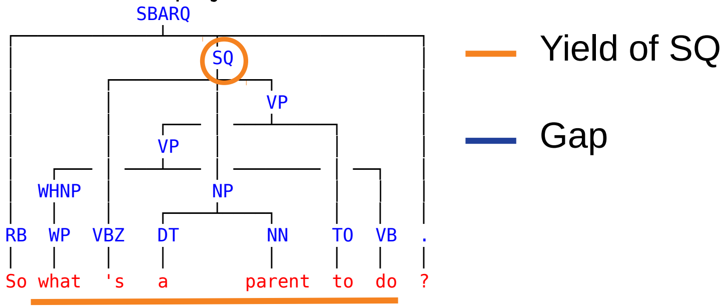
- We extend **span vectors** (Wang and Chang, 2016; Cross and Huang, 2016) to **discontinuous constituents**
- Concatenate 4 contextual vectors to represent a constituent:
  - **first** and **last** token in **yield** of constituent
  - **first** and **last** token in **gap** of constituent



- $\mathbf{r}(\mathrm{VP}, \{\mathrm{what}, \mathrm{to}, \mathrm{do}\}) = [\mathbf{h}_{\mathrm{what}} ; \mathbf{h}_{\mathrm{do}} ; \mathbf{h}_{\mathrm{'s}} ; \mathbf{h}_{\mathrm{parent}}]$

Scoring System: Constituent Representations

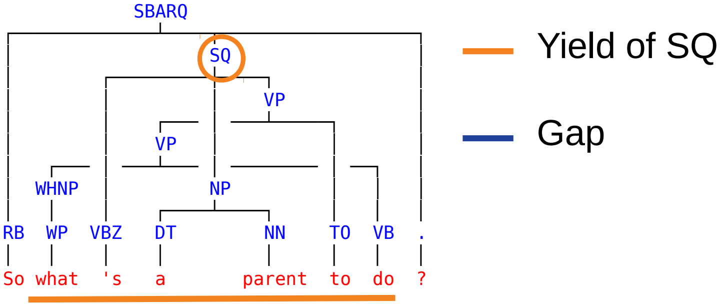
- What if my constituent is projective?



- $\mathbf{r}(\mathrm{SQ}) = [\mathbf{h}_{\mathrm{what}} ; \mathbf{h}_{\mathrm{do}} ; \mathbf{h}_{\mathrm{nil}} ; \mathbf{h}_{\mathrm{nil}}]$
- $\mathbf{h}_{\mathrm{nil}}$ is a learned parameter vector

Scoring System: Constituent Representations

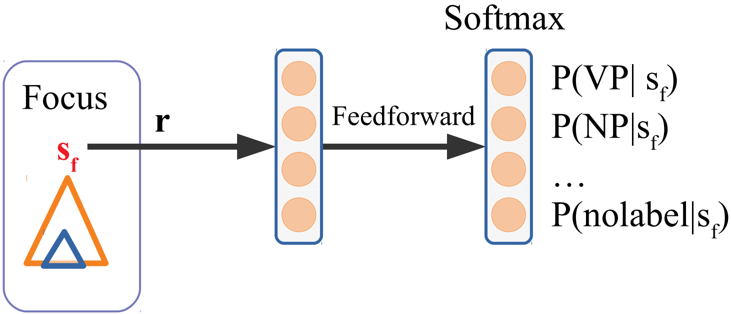
- What if my constituent is projective?



  - $\mathbf{r}(SQ) = [\mathbf{h}_{what} ; \mathbf{h}_{do} ; \mathbf{h}_{nil} ; \mathbf{h}_{nil}]$
  - $\mathbf{h}_{nil}$ is a learned parameter vector
- Sure, but what if my constituent has **2 gaps**?
  - ☹ Our method is not expressive enough to represent a constituent with 2 gaps
  - Constituents will have distinct representations as long as they have at most a single gap

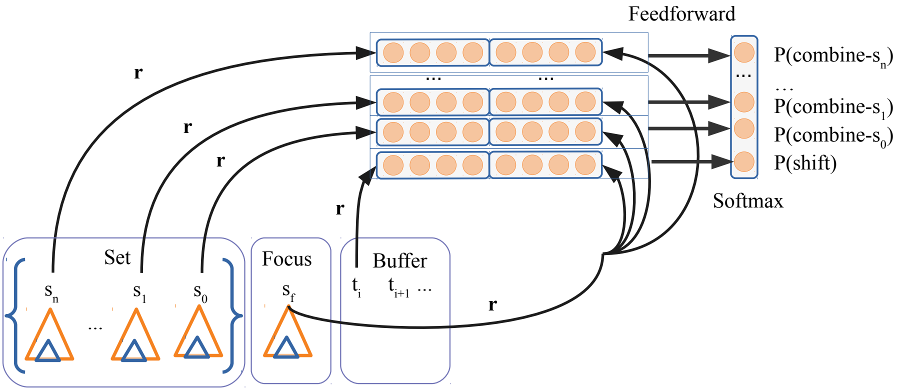
Scoring System: Scoring Labelling Actions

- Is s_f (focus item) a constituent, and if so, what is its label?



- **r**: discontinuous constituent representation function
- Feedforward: 2 hidden layers with tanh activation

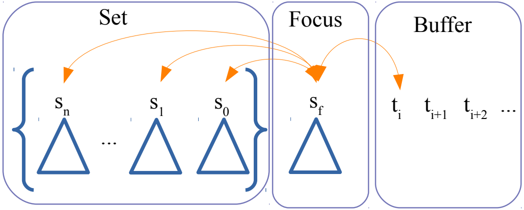
Scoring System: Scoring Structural Actions



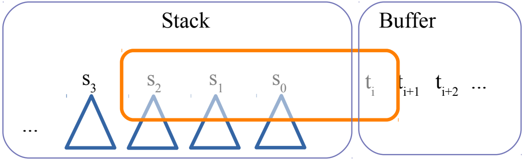
- Score independently pairwise combinations (s_k, s_f)
- Score shift with the combination $(\{i\}, s_f)$ (i: first token in buffer)
- Feedforward: 2 hidden layers with tanh activation

Scoring System

- **Set-based** system: we score every possible combine-*s* → **Global view on memory**



- vs standard **stack-based** system: extract features from **local region** of a configuration and feed them to a classifier



Experiments: Settings

Datasets:

- English: Discontinuous Penn Treebank (Evang and Kallmeyer, 2011)
  Around 20% of sentences contain a discontinuity
- German: Tiger Corpus (Brants et al., 2004), 30% of sentences with a discontinuity
- German: Negra Corpus (Skut et al., 1997)

Training:

- Supervised setting (no pretrained embeddings or external data)
- Compare:
  - **Static oracle**: teacher forcing, train on gold (configuration, action) pairs only
    - Loss = negative log likelihood of gold derivations
  - **Dynamic oracle**: sample action from predicted action distribution
    - Loss = negative log likelihood of best actions given sampled configuration

Greedy decoding in all experiments.

Experiments: Results on Development Corpora

	DPTB		Tiger		Negra	
	F1	Disc. F1	F1	Disc. F1	F1	Disc. F1
static	91.1	68.2	87.4	61.7	83.6	51.3
dynamic	91.4	70.9	87.6	62.5	84.0	54.0
Δ	**+0.3**	**+2.7**	**+0.2**	**+0.8**	**+0.4**	**+2.7**

- F1: Fscore on all constituents
- Disc. F1: Fscore computed on discontinuous constituents only

$\longrightarrow$ Dynamic oracle improves learning

Experiments: State-of-the-Art Results on Test Corpora

		English (DPTB)		German (Tiger)		German (Negra)	
Model	POS	F	Disc. F	F	Disc. F	F	Disc. F
Ours, dynamic oracle	own	**90.9**	67.3	82.5	**55.9**	**83.2**	**56.3**
Other transition-based parsers							
Coavoux et al. (2019), **gap**	own	**91.0**	**71.3**	**82.7**	**55.9**	**83.2**	54.6
Stanojević and Garrido Alhama (2017), **swap**	pred			77.0			
Stanojević and Garrido Alhama (2017), **swap**	gold			81.6		82.9	
Coavoux and Crabbé (2017), **gap**	pred			79.3			
Other methods							
Corro et al. (2017): **dependency-based**	pred	89.2					
van Cranenburgh et al. (2016), ≤ 40, **grammar-based**	own	87.0				74.8	
Versley (2016), **grammar-based**	own			79.5			

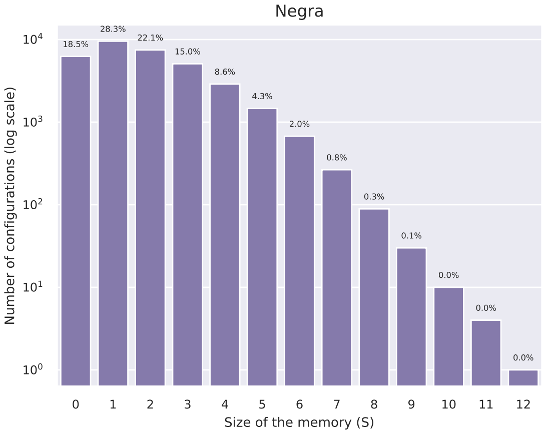
- Using fine-tuned BERT for token representation boosts results on DPTB to:
  - Test: **94.8** F1 (Disc. F1 **74.7**)
  - Dev: 95.0 F1 (Disc. F1 79.4)

Model Analysis

- Since the model needs to score every possible combinations, its complexity depends on the **size of the set**.
- Worst case: the set contains $n-1$ singletons.
- Works well under assumption that the set remains small (i.e. parsing is incremental), does it hold in practice?

Model Analysis

- Since the model needs to score every possible combinations, its complexity depends on the **size of the set**.
- Worst case: the set contains $n-1$ singletons.
- Works well under assumption that the set remains small (i.e. parsing is incremental), does it hold in practice?



The set has fewer than 8 items for more than 99% configurations.

Conclusion

- New transition system with a **set-structured memory**

- Derive any discontinuous constituency tree in exactly $4n-2$ actions

- Code release with pretrained models: `gitlab.com/mcoavoux/discoparset`

- Data release: complete wikipedia (German and English) parsed to discontinuous constituency trees
  - `https://github.com/mcoavoux/wiki_parse`
  - `http://www.llf.cnrs.fr/wikiparse/`

Conclusion

- New transition system with a **set-structured memory**
- Derive any discontinuous constituency tree in exactly $4n-2$ actions
- Code release with pretrained models: `gitlab.com/mcoavoux/discoparset`
- Data release: complete wikipedia (German and English) parsed to discontinuous constituency trees
  - `https://github.com/mcoavoux/wiki_parse`
  - `http://www.llf.cnrs.fr/wikiparse/`

Thank you for your attention!

Thanks to: Caio Corro, Giorgio Satta, Marco Damonte.

References I

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. Tiger: Linguistic interpretation of a german corpus. *Research on Language and Computation*, 2(4): 597–620, 2004. ISSN 1572-8706. doi: 10.1007/s11168-004-7431-3. URL http://dx.doi.org/10.1007/s11168-004-7431-3.

Maximin Coavoux and Benoit Crabbé. Incremental discontinuous phrase structure parsing with the gap transition. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1259–1270, Valencia, Spain, April 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E17-1118.

Maximin Coavoux, Benoît Crabbé, and Shay B. Cohen. Unlexicalized transition-based discontinuous constituency parsing. *CoRR*, abs/1902.08912v1, 2019. URL http://arxiv.org/abs/1902.08912v1.

Caio Corro, Joseph Le Roux, and Mathieu Lacroix. Efficient discontinuous phrase-structure parsing via the generalized maximum spanning arborescence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1645–1655, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1172.

James Cross and Liang Huang. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1–11, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1001.

Kilian Evang and Laura Kallmeyer. PLCFRS parsing of english discontinuous constituents. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 104–116, Dublin, Ireland, October 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W11-2913.

Wolfgang Maier. Discontinuous incremental shift-reduce parsing. 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 1202–1212, Beijing, China, July 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P15-1116.

References II

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 88–95, Washington, DC, USA, March 1997. Association for Computational Linguistics. doi: 10.3115/974557.974571. URL http://www.aclweb.org/anthology/A97-1014.

Miloš Stanojević and Raquel Garrido Alhama. Neural discontinuous constituency parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1667–1677, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1174.

Andreas van Cranenburgh, Remko Scha, and Rens Bod. Data-oriented parsing with discontinuous constituents and function tags. *Journal of Language Modelling*, 4(1):57–111, 2016. URL http://dx.doi.org/10.15398/jlm.v4i1.100.

Yannick Versley. Discontinuity re^2-visited: A minimalist approach to pseudoprojective constituent parsing. In *Proceedings of the Workshop on Discontinuous Structures in Natural Language Processing*, pages 58–69, San Diego, California, June 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W16-0907.

Wenhui Wang and Baobao Chang. Graph-based dependency parsing with bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2306–2315, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1218. URL https://www.aclweb.org/anthology/P16-1218.