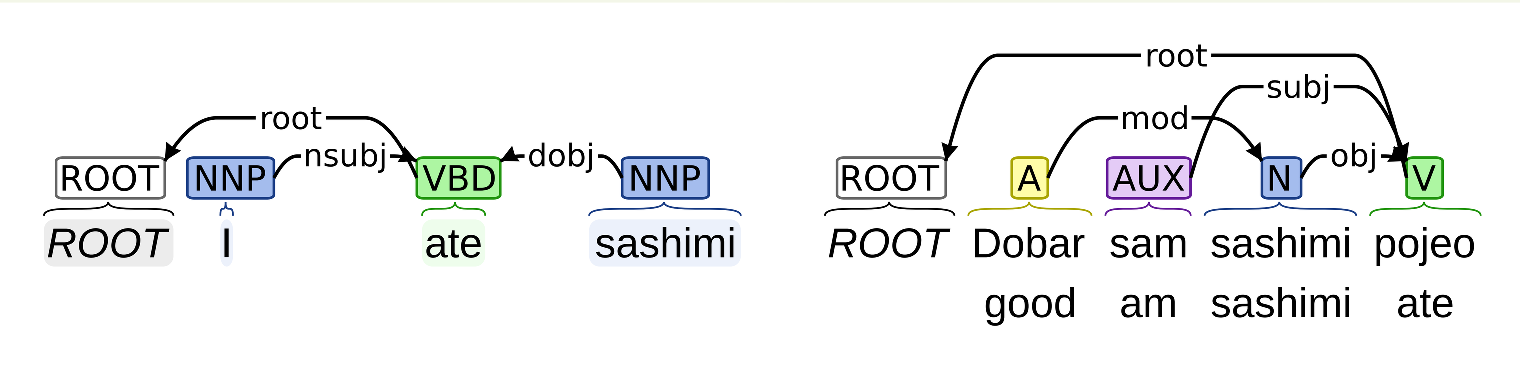
Transition-based Dependency Parsing Using Recursive Neural Networks

Pontus Stenetorp* | National Institute of Informatics, Tokyo, Japan | pontus@stenetorp.se
* Currently at the University of Tokyo.

Why Parsing?

- Who did what to whom?
- Required for natural language understanding (most likely).
- Syntactic/semantic connection.
- Key task in the Natural Language Processing (NLP) community.

Syntacto-semantic Dependencies



(a) English. (b) Croatian ("I ate good sashimi").

Figure: Example sentences and their dependency trees.

- A brief introduction:
  - Focus on words and their relations.
  - Flexible enough for language phenomena such as non-projectives.
  - Corpora available for a large set of languages.
  - Simply a connected labeled directed graph.
- Terminology:
  - A *dependent* is attached to a *head*.
  - Each head-dependent relation has a *dependency type*.

Previous Work and Conceptual Problems

- Vector composition and constituency parsing:
  - Recursive Neural Network (RNN) model (Socher et al. 2010).
  - Works within the constituency tree.
  - Produces phrase representations and constituency trees.
- "Vanilla" RNN approach not applicable to dependency trees:
  - Different number of parents for non-sources.
  - Can not handle non-projectives.

Transition-based Dependency Parsing

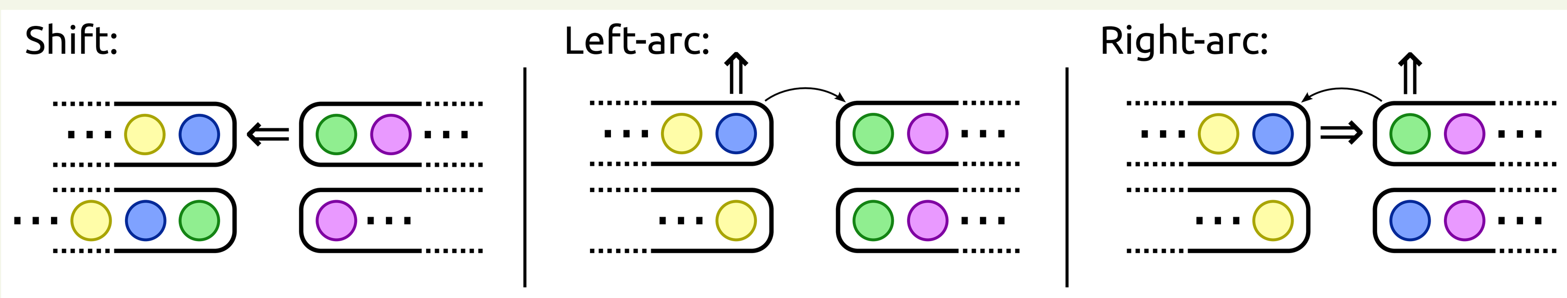


Figure: *Arc-Standard transitions.*

- Incremental state machine:
  - A stack and a buffer.
  - Transitions operating on the stack/buffer.
  - Efficient and arguably cognitively plausible.
- Variants:
  - Projective: Arc-Standard and Arc-Eager.
  - Non-projective: Swap.
  - And more...
- For this poster we will focus on the *Arc-Standard* variant.

A Compositional Vector Framework

- Vector representations as opposed to words/trees.
- Compose the representations and predict a transition.
- Compositional and non-compositional transitions.
- Results in a Transition Directed Acyclic Graph (DAG).

Model

- *Arc-Standard* algorithm cast in our framework.
- Replace the head with the composition of the head/dependent.
- Greedy search and global weight updates.
- Observes the top 3 representations of the stack/buffer (*horizon*).
- Single composition matrix $\boldsymbol{W_C} \in \mathbb{R}^{n \times 6n}$.
- SoftMax weights $\boldsymbol{W_S} \in \mathbb{R}^{3 \times n}$.
- Only word representations as input $\boldsymbol{a_i} \in \mathbb{R}^{n}$.

Composition and Parsing Example

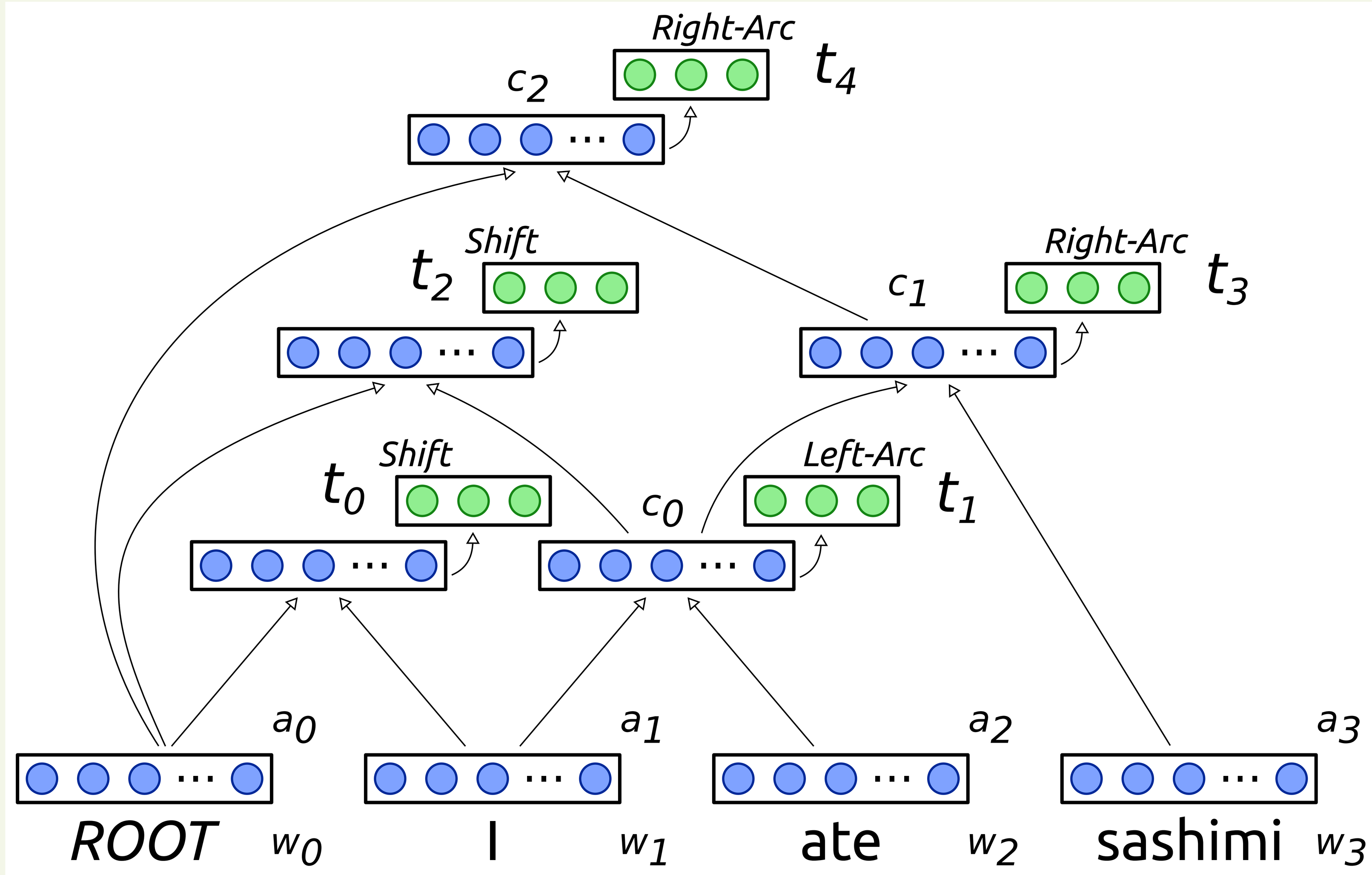


Figure: Transition DAG for our English example sentence.

	Transition	Stack	Buffer	Arcs	Compositions
		[$ROOT_{w_0}$]	[I_{w_1}, ate_{w_2}, . . .]		
t_0	*Shift* ⇒	[$ROOT_{w_0}$, I_{w_1}]	[ate_{w_2}, $sashimi_{w_3}$]		
t_1	*Left-Arc* ⇒	[$ROOT_{w_0}$]	[ate_{w_2}, $sashimi_{w_3}$]	I → ate	$c_0 = p([a_1; a_2])$
t_2	*Shift* ⇒	[$ROOT_{w_0}$, ate_{w_2}]	[$sashimi_{w_3}$]		
t_3	*Right-Arc* ⇒	[$ROOT_{w_0}$]	[ate_{w_2}]	sashimi → ate	$c_1 = p([c_0; a_3])$
t_4	*Right-Arc* ⇒	[]	[$ROOT_{w_0}$]	ate → *ROOT*	$c_2 = p([a_0; c_1])$

Table: Oracle transitions for our English example sentence.

- Examples for non-projectives and other algorithms can be found in the paper.

Training

- Generate gold Transition DAGs using oracle transitions.
- 200-dimensional word representations by Turian et al. (2010).
- Diagonal version of AdaGrad for optimisation.

Quantitative Results

	Model	UAS
(1)	*This work*	**86.25%**
(2)	Comparable Feature-based System	**88.06%**
(3)	Shared-task Top System	***92.45%***

Table: Unlabeled Attachment Score (UAS) for our model.

- CoNLL 2008 Shared Task Data Set.

Qualitative Results

(a)	a financial crisis
1st	a cash crunch
2nd	a bear market
(b)	hammer out their own plan
1st	work out their own compromise
2nd	enact the cut this year
(c)	to run their computerized trading strategies
1st	to determine buy and sell orders
2nd	to pick up more shares today
(d)	from $ 142.7 million , or 78 cents a share
1st	from $ 367.1 million , or $ 2.05 a share
2nd	from the sale of its First Chicago Investment Advisors unit

Table: Nearest neighbour phrases.

- Parse the development set, representations for each phrase.
- Query phrase and its two nearest neighbours.

Conclusions and Future Work

- Conclusions:
  - First Deep Learning-based approach to dependency parsing.
  - Performs within **2%** UAS to a comparable feature-based model.
  - Produces similar phrase representations as Socher et al. (2010).
- Future work:
  - Compositional vector parsing for "any" language.
  - "Horizon-free" dependency parsing.
  - Untied weights and other improvements to reach for the state-of-the-art.

Deep Learning Workshop, The 2013 Conference on Neural Information Processing Systems (NIPS), Lake Tahoe, Nevada, USA, December 9th 2013