# Parsing with Compositional Vector Grammars

Richard Socher
John Bauer
Christopher Manning
Andrew Y Ng

Hrishikesh Terdalkar, 14111265
{ hrishirt } @ iitk

28 August 2015

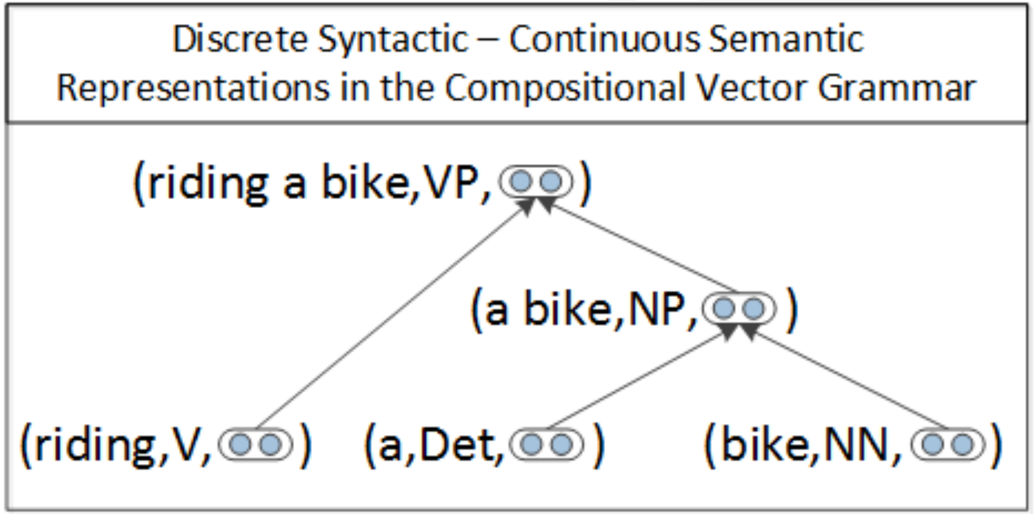
# Overview

- traditional representation using NP and VP **does not** capture the full syntactic nor semantic richness of linguistic phrases
- lexicalizing phrases or splitting categories only partly address problem at cost of huge feature spaces and sparseness.
- introduction of **Compositional Vector Grammar** (CVG), which combines PCFGs with a syntactically untied RNN that learns syntactico-semantic. compositional vector representations
- CVG learns a soft notion of head words and improves performance on the types of ambiguities that require semantic information such as PP attachments

# CVG Approach

- **Compositional Vector Grammar** Parser (CVG) for structure prediction
- the model addresses the problem of representing phrases and categories, jointly learning how to parse and how to represent phrases as both discrete categories and continuous vectors (CVG Tree Example)
- combine the advantages of standard **probabilistic context free grammars** (PCFG) with those of **recursive neural networks** (RNNs)
  - **PCFG** can capture discrete categorization of phrases into NP or PP
  - **RNN** can capture fine-grained syntactic and compositional-semantic information on phrases and words
- can help in cases where **syntactic ambiguity** can <u>only</u> be resolved with the help of **semantic information**
  - *They ate udon* ***with forks*** vs. *They ate udon* ***with chicken***

# CVG Tree Example




CVG tree with (category,vector) representations at each node.

Vectors for nonterminals are computed via a new type of RNN which is conditioned on syntactic categories from a PCFG

# CVG Approach (contd.)

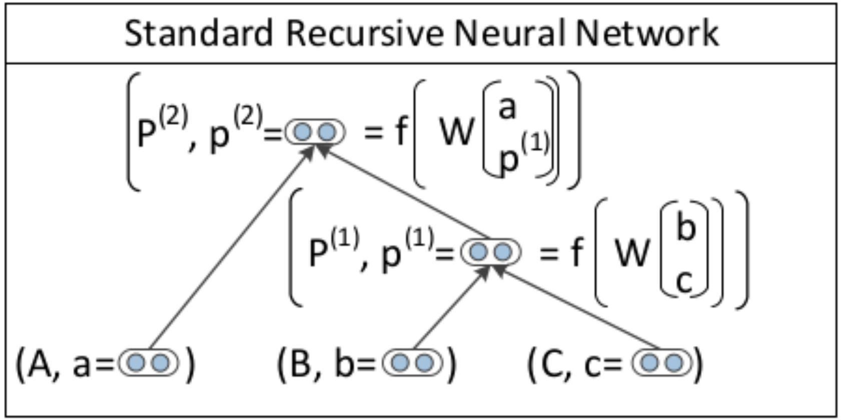
- ❖ previous RNN-based parsers used the same (tied) weights at all nodes to compute the vector representing a constituent
- ❖ hard to optimize since the parameters form a very deep neural network.
- ❖ CVG approach generalizes the fully tied RNN to one with syntactically untied weights, weights at each node are *conditionally dependent* on the categories of the child constituents.
- ❖ allows different composition functions when combining different types of phrases and is shown to result in a large improvement in parsing accuracy
- ❖ **compositional distributed representation** allows a CVG parser to make accurate parsing decisions and capture similarities between phrases and sentences
- ❖ Any PCFG-based parser can be improved with an RNN.
  - ➢ simplified version of the Stanford Parser used here as base PCFG

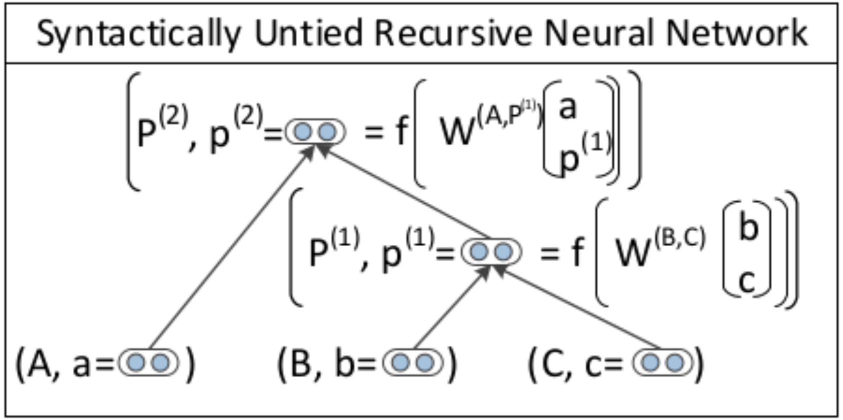
# Recursive Neural Networks

## **standard** vs. **syntactically untied**

Tree with a **simple RNN**: same weight matrix is replicated and used to compute all non-terminal node representations. Leaf nodes are n-dimensional vector representations of words.

A **syntactically untied RNN** in which the function to compute a parent vector depends on syntactic categories of its children which are assumed to be given.



# Compositional Vector Grammar (CVG)

- Word Vector Representations
  - a sentence $S$ as an ordered list of (word,vector) pairs: $x = ((w_1, a_{w1}), \ldots, (w_m, a_{wm}))$
- Max-Margin Training Objective for CVGs
  - set of all possible trees for a given sentence $x_i$ is defined as $Y(x_i)$ and the correct tree for a sentence is $y_i$
  - to minimize this objective, the score of the correct tree $y_i$ is increased and the score of the highest scoring incorrect tree $y'$ is decreased
- Scoring Trees with CVGs
  - define the word representations as (vector, POS) pairs: $((a, A), (b, B), (c, C))$
  - standard RNN essentially ignores all POS tags and syntactic categories and each non-terminal node is associated with the same neural network
  - the CVG uses a syntactically untied RNN (SU-RNN) which has a set of such weights. size of this set depends on the number of sibling category combinations in the PCFG

# Compositional Vector Grammar (CVG)

- Parsing with CVGs
  - goodness of a tree is measured in terms of its score and the CVG score of a complete tree is the sum of the scores at each node
  - the SU-RNN rule score computation at each node still only has access to its child vectors, not the whole tree or other global features
  - allows the second pass to be very fast
- Training SU-RNNs
  - full CVG model is trained in two stages
  - first the base PCFG is trained and its top trees are cached and then used for training the SU-RNN conditioned on the PCFG
  - SU-RNN is trained using Max-Margin Training objective and scores as exemplified earlier.
- Subgradient Methods and AdaGrad
  - the learning rate is adapting differently for each parameter and rare parameters get larger updates than frequently occurring parameters
- Initializing of Weight Matrices
  - in absense of any knowledge, for combining two vectors is to average them instead of performing a completely random projection
  - $W^{(AB)} [\, a, b, 1] = W^{(A)}a + W^{(B)}b + bias$

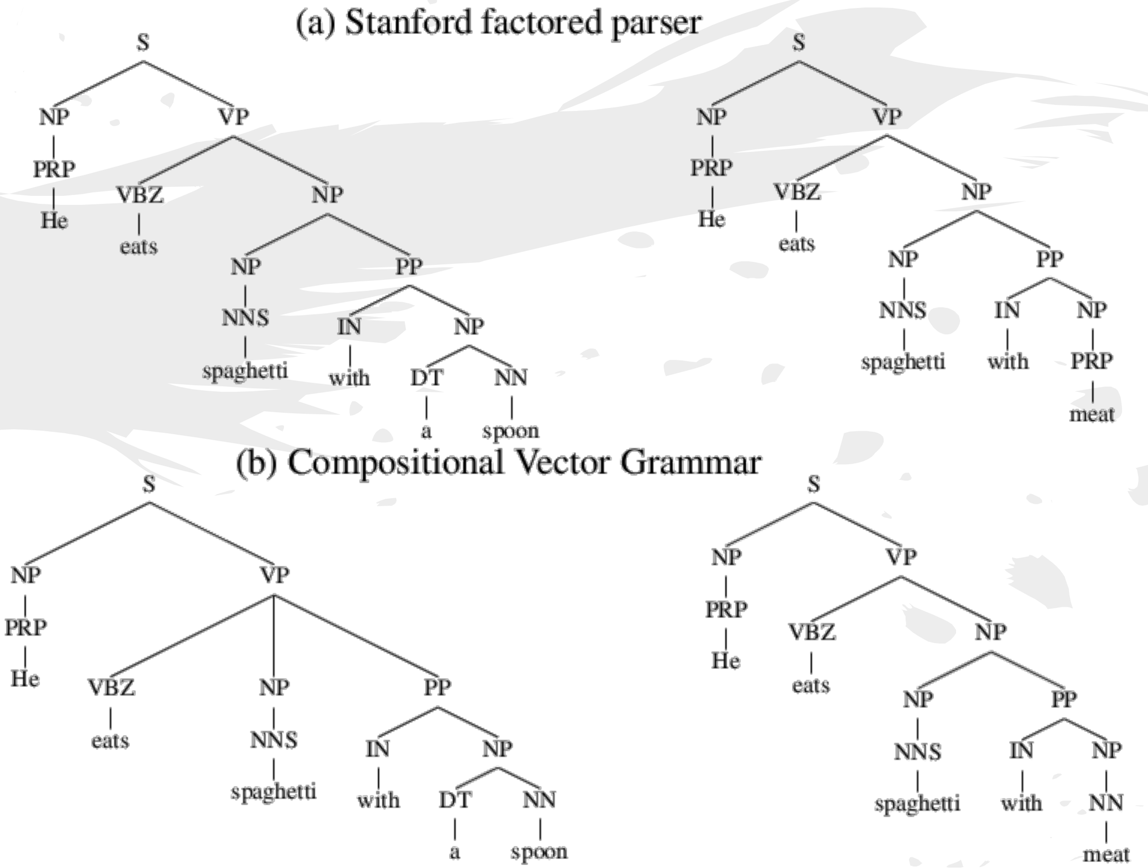
**Comparison of parsers with richer state representations on the WSJ. The last line is the self-trained re-ranked Charniak parser.**

| Parser | dev (all) | test ≤ 40 | test (all) |
|---|---|---|---|
| Stanford PCFG | 85.8 | 86.2 | 85.5 |
| Stanford Factored | 87.4 | 87.2 | 86.6 |
| Factored PCFGs | 89.7 | 90.1 | 89.4 |
| Collins | | | 87.7 |
| SSN (Henderson) | | | 89.4 |
| Berkeley Parser | | | 90.1 |
| **CVG (RNN)** | 85.7 | 85.1 | 85.0 |
| **CVG (SU-RNN)** | **91.2** | **91.1** | **90.4** |
| Charniak-SelfTrain | | | 91.0 |
| Charniak-RS | | | 92.1 |

# Analysis of Error Types: Detailed Comparison of different parsers

| Error Type | Stanford | CVG | Berkley | Char-RS |
|---|---|---|---|---|
| PP Attach | 1.02 | 0.79 | 0.82 | **0.60** |
| Clause Attach | 0.64 | 0.43 | 0.50 | **0.38** |
| Diff Label | 0.40 | **0.29** | **0.29** | 0.31 |
| Mod Attach | 0.37 | 0.27 | 0.27 | **0.25** |
| NP Attach | 0.44 | 0.31 | 0.27 | **0.25** |
| Co-ord | 0.39 | 0.32 | 0.38 | **0.23** |
| 1-Word Span | 0.48 | 0.31 | 0.28 | **0.20** |
| Unary | 0.35 | 0.22 | 0.24 | **0.14** |
| NP Int | 0.28 | 0.19 | 0.18 | **0.14** |
| Other | 0.62 | **0.41** | **0.41** | 0.50 |

Test sentences of semantic transfer for PP attachments.
**CVG** was able to transfer semantic word knowledge from two related training sentences.
In contrast, **Stanford parser** could not distinguish the PP attachments based on the word semantics.



# Conclusion

- parsing model that combines the speed of small-state PCFGs with semantic richness of neural word representations and compositional phrase vectors
- compositional vectors are learned with a new *syntactically untied recursive neural network (RNN)*
- linguistically more plausible since it chooses different composition functions for a parent node based the syntactic categories of its children
- **CVG** obtains 90.44% labeled F1 on the full WSJ test set and is 20% faster than the previous Stanford parser.
- not the best model, but fast
- huge number of parameters:

  $d * vocab + 2d * d * (n_{comp}) + d * class + d$
- can't make the *standard* RNN perform better than the PCFG, but a very creative modification to the standard RNN

# References

- Richard Socher, John Bauer, Christopher Manning and Andrew Y Ng, **Parsing with Compositional Vector Grammars**, In Proceedings of *ACL Conference 2013*.