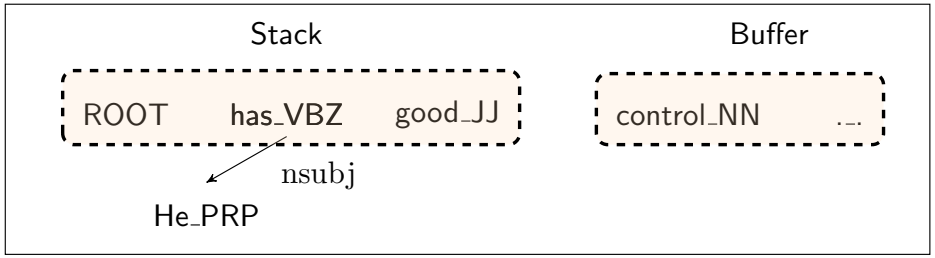
Traditional Features



binary, sparse
dim =10^6 ~ 10^7

Indicator features

$$s_2.w = \text{has} \wedge s_2.t = \text{VBZ}$$
$$s_1.w = \text{good} \wedge s_1.t = \text{JJ} \wedge b_1.w = \text{control}$$
$$lc(s_2).t = \text{PRP} \wedge s_2.t = \text{VBZ} \wedge s_1.t = \text{JJ}$$
$$lc(s_2).w = \text{He} \wedge lc(s_2).l = \text{nsubj} \wedge s_2.w = \text{has}$$

leftmost child

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Indicator Features Revisited

$$s_2.w = \text{has} \wedge s_2.t = \text{VBZ}$$
$$s_1.w = \text{good} \wedge s_1.t = \text{JJ} \wedge b_1.w = \text{control}$$
$$lc(s_2).t = \text{PRP} \wedge s_2.t = \text{VBZ} \wedge s_1.t = \text{JJ}$$
$$lc(s_2).w = \text{He} \wedge lc(s_2).l = \text{nsubj} \wedge s_2.w = \text{has}$$

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Indicator Features Revisited

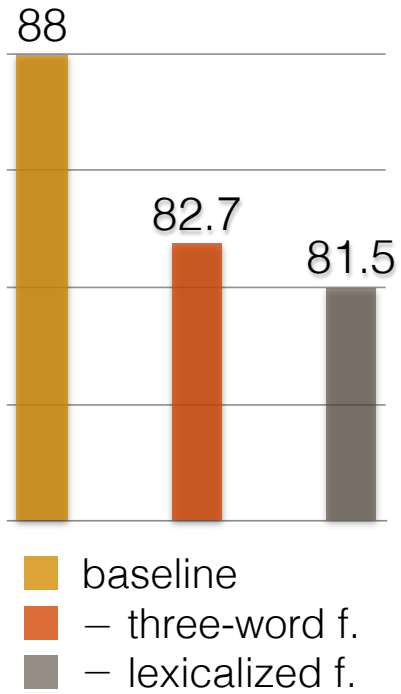
- Problem #1: sparse
  - lexicalized features
  - high-order interaction features

$$s_2.w = \text{has} \wedge s_2.t = \text{VBZ}$$
$$s_1.w = \text{good} \wedge s_1.t = \text{JJ} \wedge b_1.w = \text{control}$$
$$lc(s_2).t = \text{PRP} \wedge s_2.t = \text{VBZ} \wedge s_1.t = \text{JJ}$$
$$lc(s_2).w = \text{He} \wedge lc(s_2).l = \text{nsubj} \wedge s_2.w = \text{has}$$

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Indicator Features Revisited

- Problem #1: sparse
  - lexicalized features
  - high-order interaction features



$$s_2.w = \text{has} \wedge s_2.t = \text{VBZ}$$
$$s_1.w = \text{good} \wedge s_1.t = \text{JJ} \wedge b_1.w = \text{control}$$
$$lc(s_2).t = \text{PRP} \wedge s_2.t = \text{VBZ} \wedge s_1.t = \text{JJ}$$
$$lc(s_2).w = \text{He} \wedge lc(s_2).l = \text{nsubj} \wedge s_2.w = \text{has}$$

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Indicator Features Revisited

- Problem #1: sparse

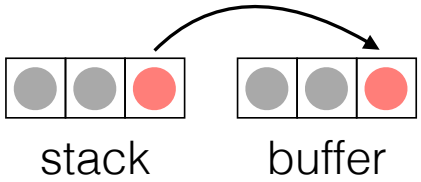
$$s_2.w = \text{has} \wedge s_2.t = \text{VBZ}$$
$$s_1.w = \text{good} \wedge s_1.t = \text{JJ} \wedge b_1.w = \text{control}$$
$$lc(s_2).t = \text{PRP} \wedge s_2.t = \text{VBZ} \wedge s_1.t = \text{JJ}$$
$$lc(s_2).w = \text{He} \wedge lc(s_2).l = \text{nsubj} \wedge s_2.w = \text{has}$$

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Indicator Features Revisited

- Problem #1: sparse
- Problem #2: incomplete

Unavoidable in hand-crafted feature templates.



~~RIGHT-ARC~~

$$s_2.w = \text{has} \wedge s_2.t = \text{VBZ}$$
$$s_1.w = \text{good} \wedge s_1.t = \text{JJ} \wedge b_1.w = \text{control}$$
$$lc(s_2).t = \text{PRP} \wedge s_2.t = \text{VBZ} \wedge s_1.t = \text{JJ}$$
$$lc(s_2).w = \text{He} \wedge lc(s_2).l = \text{nsubj} \wedge s_2.w = \text{has}$$

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Indicator Features Revisited

- Problem #1: sparse
- Problem #2: incomplete

$$s_2.w = \text{has} \wedge s_2.t = \text{VBZ}$$
$$s_1.w = \text{good} \wedge s_1.t = \text{JJ} \wedge b_1.w = \text{control}$$
$$lc(s_2).t = \text{PRP} \wedge s_2.t = \text{VBZ} \wedge s_1.t = \text{JJ}$$
$$lc(s_2).w = \text{He} \wedge lc(s_2).l = \text{nsubj} \wedge s_2.w = \text{has}$$

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Indicator Features Revisited

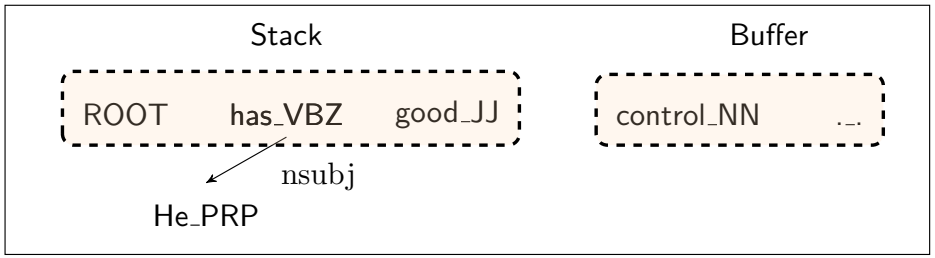
- Problem #1: sparse
- Problem #2: incomplete
- Problem #3: computationally expensive

More than 95% of parsing time is consumed by feature computation.

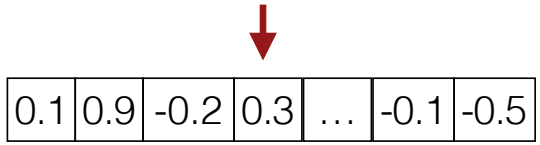
$$s_2.w = \text{has} \wedge s_2.t = \text{VBZ}$$
$$s_1.w = \text{good} \wedge s_1.t = \text{JJ} \wedge b_1.w = \text{control}$$
$$lc(s_2).t = \text{PRP} \wedge s_2.t = \text{VBZ} \wedge s_1.t = \text{JJ}$$
$$lc(s_2).w = \text{He} \wedge lc(s_2).l = \text{nsubj} \wedge s_2.w = \text{has}$$

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Indicator Features Revisited



dense
dim = 200

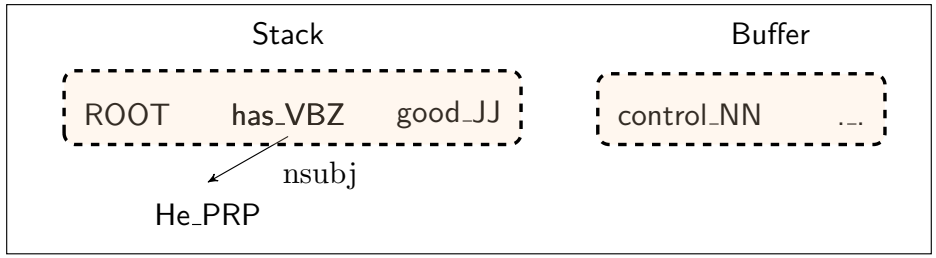


Our Solution: Neural Networks!

Learn a **dense** and **compact** feature representation

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

The Challenge



dense
dim = 200

- How to encode all the available information?
- How to model high-order features?

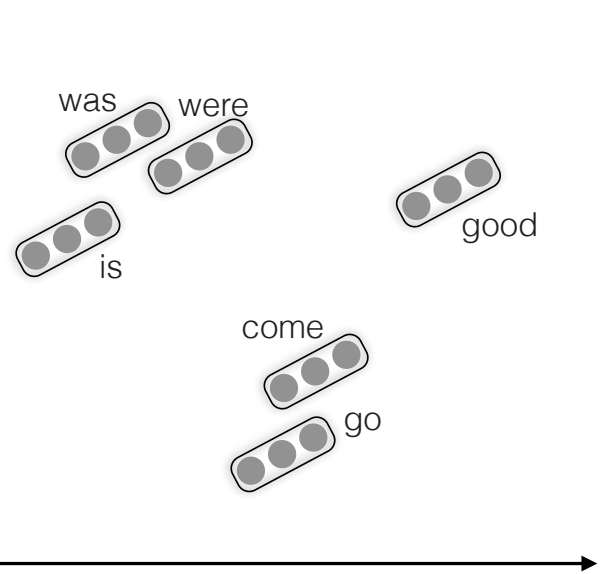
Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Distributed Representations

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Distributed Representations

- We represent each word as a d-dimensional dense vector (i.e., word embeddings).
  - Similar words expect to have close vectors.



Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Distributed Representations

- We represent each word as a d-dimensional dense vector (i.e., word embeddings).
  - Similar words expect to have close vectors.
- Meanwhile, part-of-speech tags and dependency labels are also represented as d-dimensional vectors.
  - POS and dependency embeddings.
  - The smaller discrete sets also exhibit many semantical similarities.

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Distributed Representations

- We represent each word as a d-dimensional dense vector (i.e., word embeddings).
  - Similar words expect to have close vectors.
- Meanwhile, part-of-speech tags and dependency labels are also represented as d-dimensional vectors.
  - POS and dependency embeddings.
  - The smaller discrete sets also exhibit many semantical similarities.

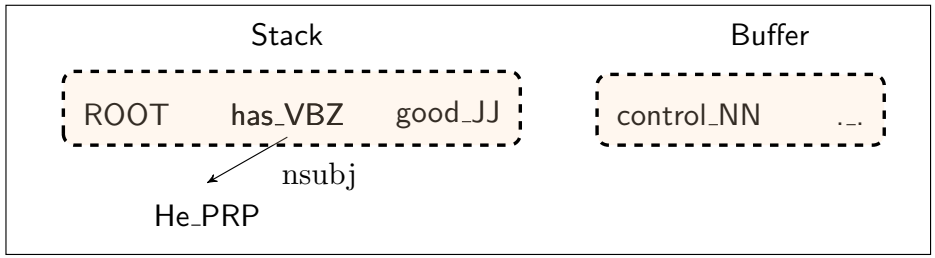
NNS (plural noun) should be close to NN (singular noun).

num (numerical modifier) should be close to amod (adjective modifier).

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Extracting Tokens from Configuration

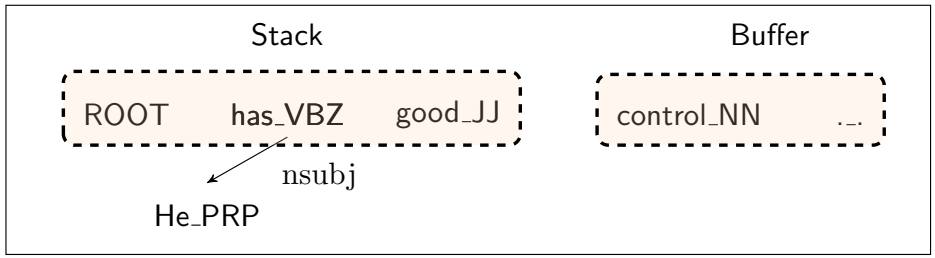
- We extract a set of tokens based on the positions:



Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Extracting Tokens from Configuration

- We extract a set of tokens based on the positions:

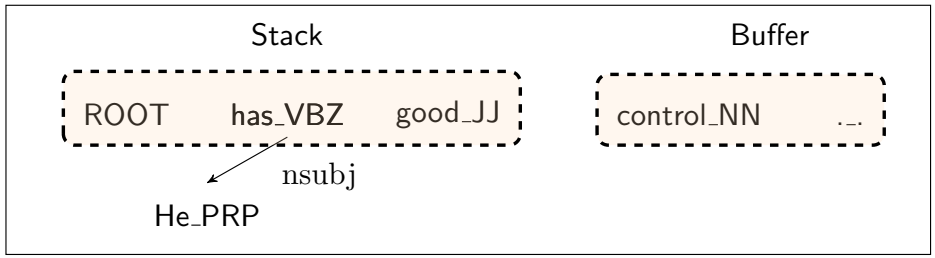


s_1
s_2
b_1
$lc(s_1)$
$rc(s_1)$
$lc(s_2)$
$rc(s_2)$
...

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Extracting Tokens from Configuration

- We extract a set of tokens based on the positions:



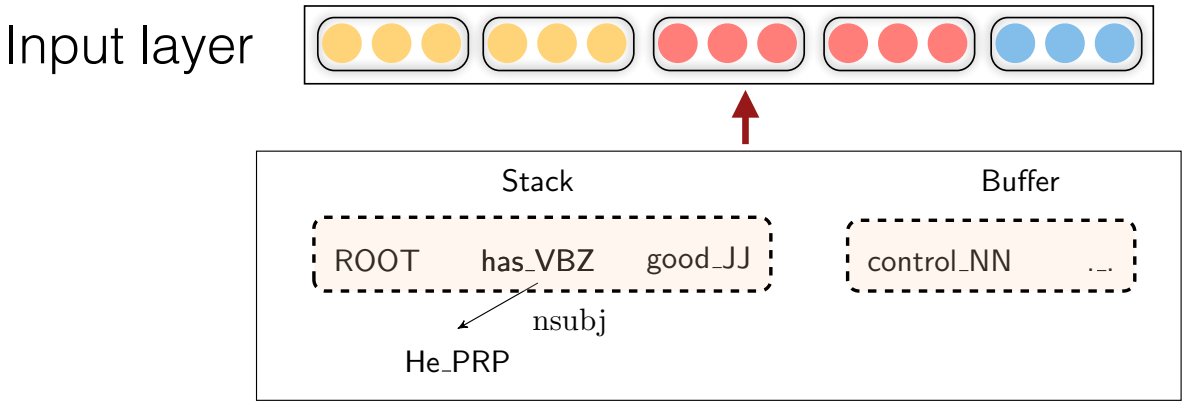
	word	POS	dep.
s_1	good	JJ	
s_2	has	VBZ	
b_1	control	NN	
$lc(s_1)$	∅	∅	∅
$rc(s_1)$	∅	∅	∅
$lc(s_2)$	He	PRP	nsubj
$rc(s_2)$	∅	∅	∅
…	…	…	…

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Model Architecture

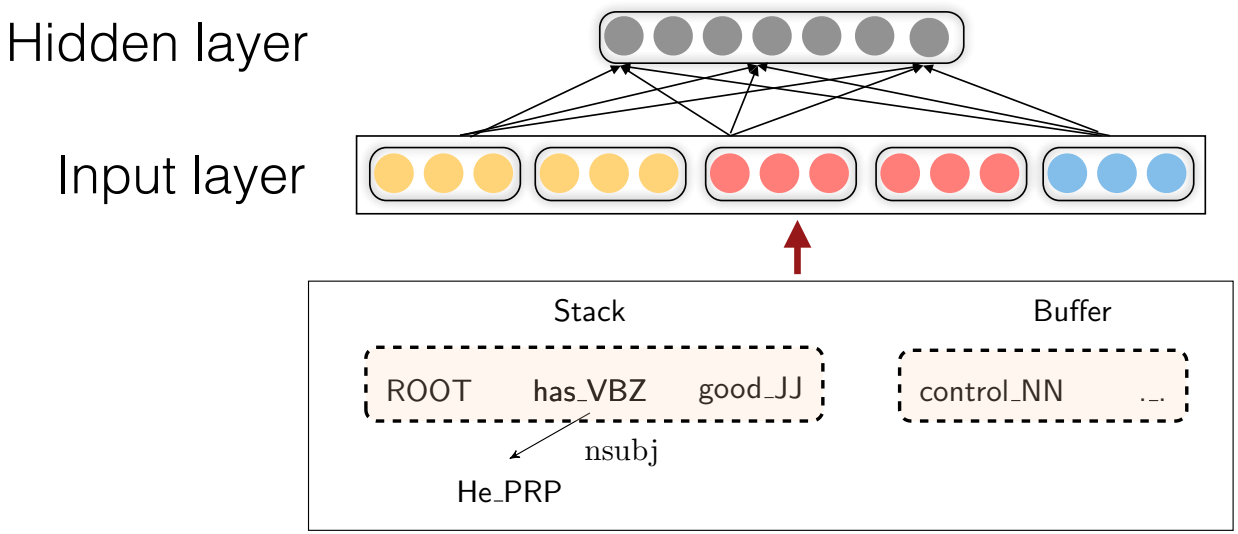
Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Model Architecture



Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

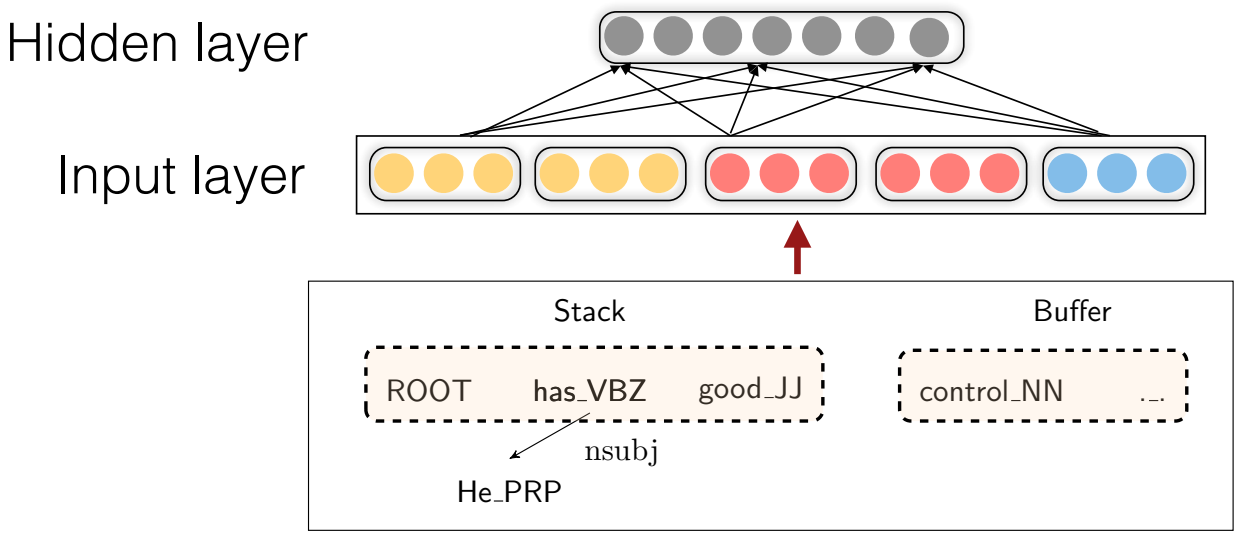
Model Architecture



Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

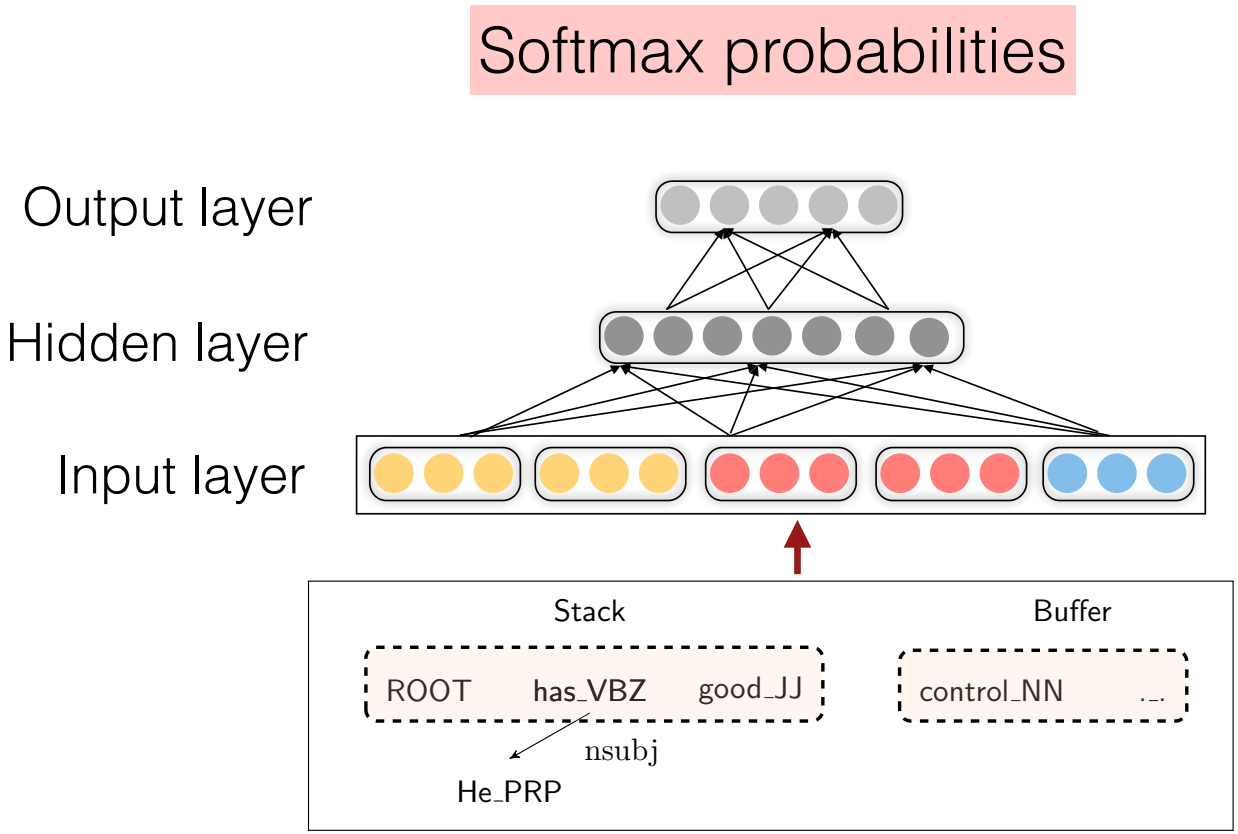
Model Architecture

Cube activation function: $g(x) = x^3$



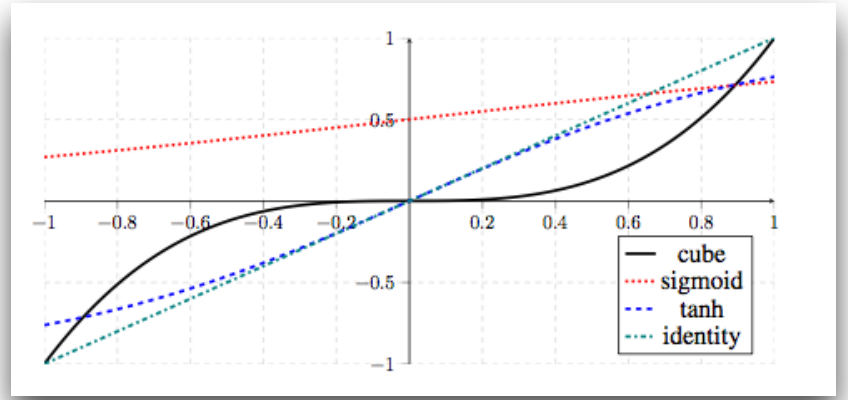
Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Model Architecture



Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Cube Activation Function



$$g(w_1x_1 + \ldots + w_mx_m + b) =$$
$$\sum_{i,j,k}(w_iw_jw_k)x_ix_jx_k + \sum_{i,j} b(w_iw_j)x_ix_j \ldots$$

Better capture the **interaction** terms!

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Training

- Generating training examples using a fixed oracle.
- **Training objective**: cross entropy loss
- Back-propagation to all embeddings.
- **Initialization**:
  - Word embeddings from pre-trained word vectors.
  - Random initialization for others.

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Indicator vs. Dense Features

- Problem #1: sparse

  Distributed representations can capture similarities.

- Problem #2: incomplete

  We don't need to enumerate the combinations.
  Cube non-linearity can learn combinations automatically.

- Problem #3: computationally expensive

  String concatenation + look-up in a big table $\Longrightarrow$ matrix operations. Pre-computation trick can speed up.

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Experimental Setup

- **Datasets**
  - English Penn Treebank (PTB)
  - Chinese Penn Treebank (CTB)
- **Representations**
  - CoNLL representations (CD) for PTB and CTB
  - Stanford Dependencies V3.3.0 (SD) for PTB
- **Part-of-speech tags**:
  - Stanford POS tagger for PTB (97.3% accuracy)
  - Gold tags for CTB

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Details

- Embedding size = 50
- Hidden size = 200
- Use mini-batched AdaGrad for optimization ($\alpha = 0.01$)
- Use 0.5 dropout on hidden layer.

- Pre-trained word vectors:
  - C & W for English
  - Word2vec for Chinese

- We use a rich set of 18 tokens from the configuration.

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

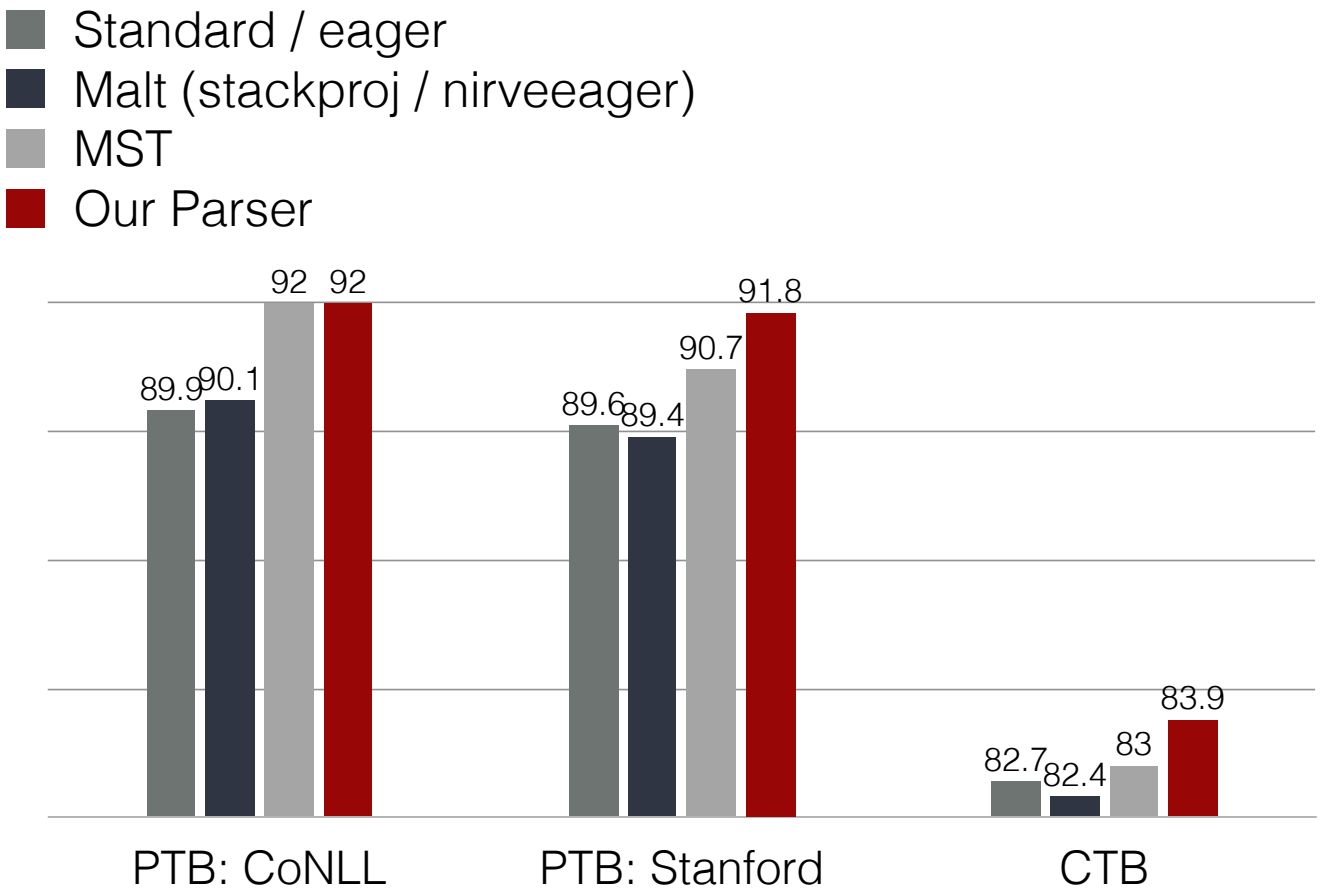
(Nivre et al 2006, McDonald and Pereira 2006)

Baselines

- **Standard / eager**: our own implemented perceptron-based greedy parsers using arc-standard or arc-eager system, with a rich feature set from (Zhang and Nivre, 2011).
- **MaltParser**
  - two algorithms **stackproj** and **nivreeager**.
- **MSTParser**

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Unlabeled Attachment Score (UAS)

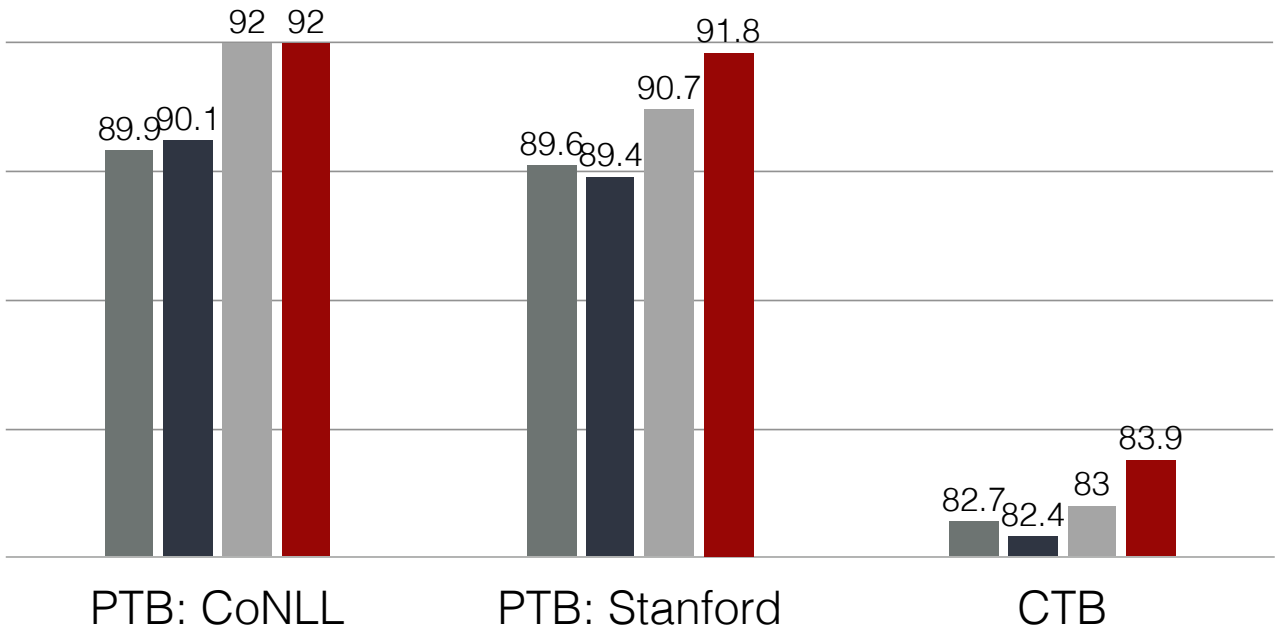


Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Unlabeled Attachment Score (UAS)

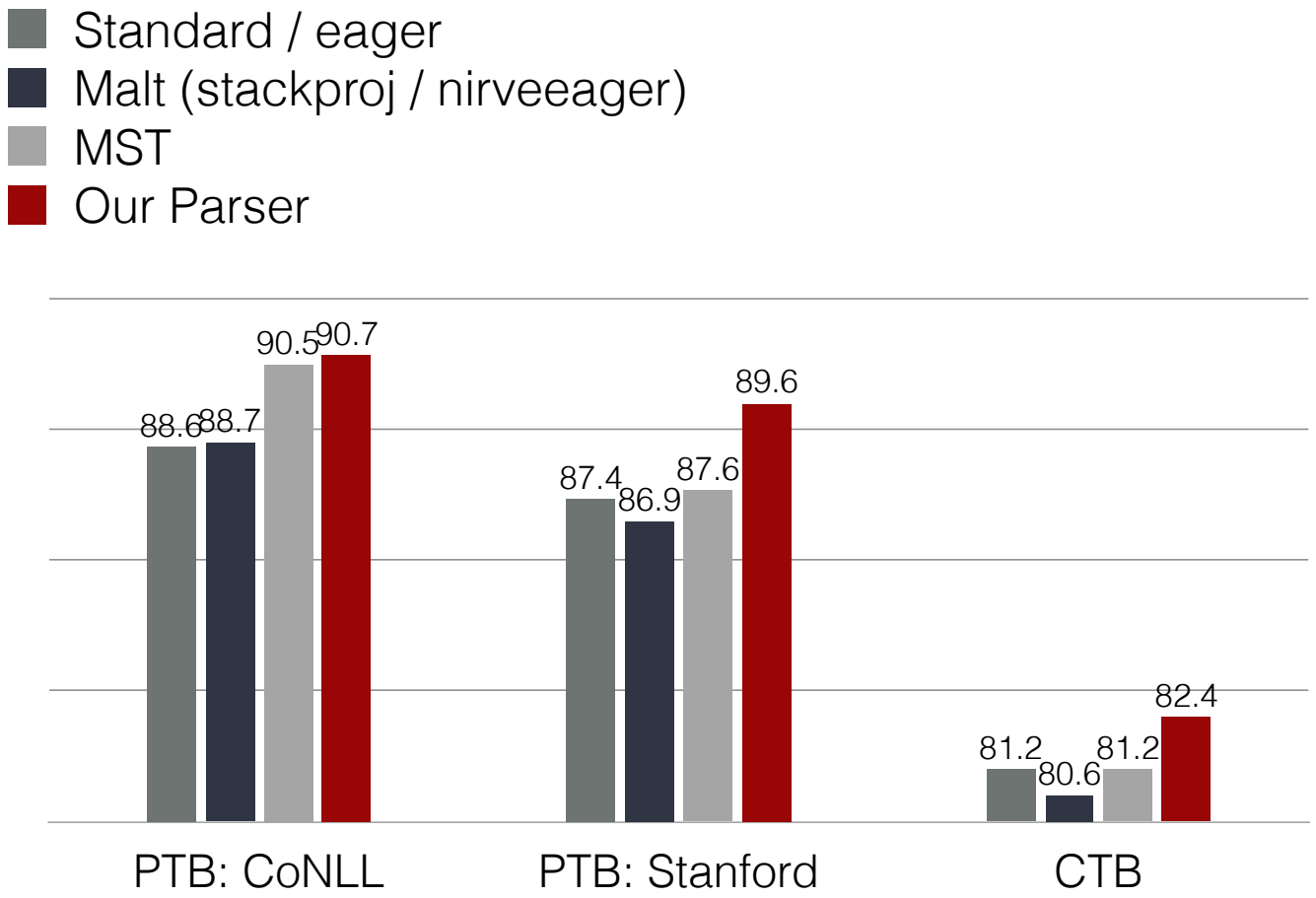
- Standard / eager
- Malt (stackproj / nirveeager)
- MST
- Our Parser

Compared with greedy parsers,
PTB: > 2.0%
CTB: >1.2%



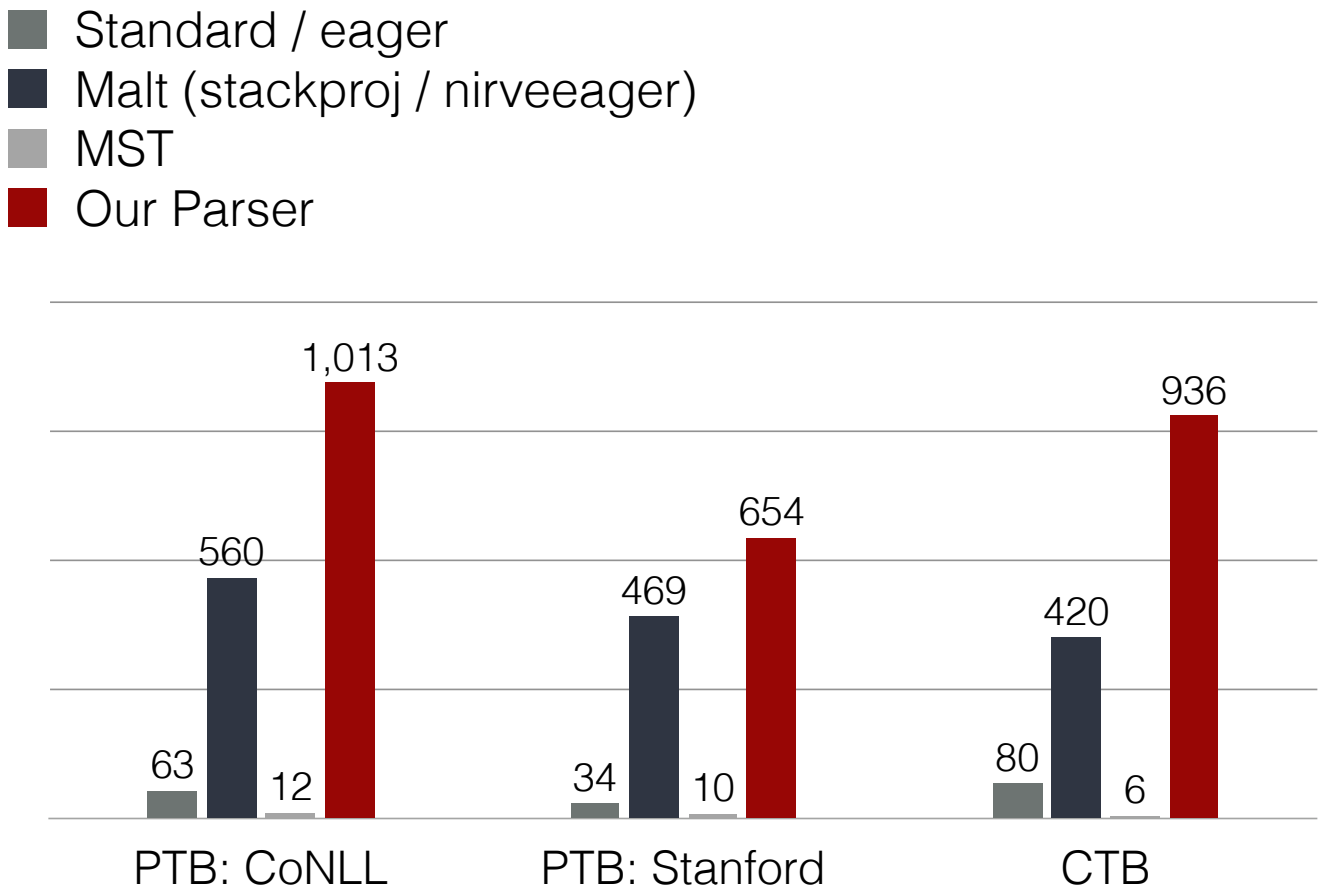
Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Labeled Attachment Score (LAS)



Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Parsing Speed (sent/s)



Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

(Zhang and Nirve 2011, Martins et al 2013)

Our Work

- A neural network based dependency parser!

Parsing on English Penn Treebank (§23):

		Unlabeled attachment score (UAS)	sent / s
Transition -based	MaltParser (greedy)	89.9	560
	Our Parser (greedy)	92.0 (+2.1)	1013 (×1.8)
	Zpar: beam = 64	92.9*	29*
Graph -based	MSTParser	92.0	12
	TurboParser	93.1*	31*

Danqi Chen and Christopher Manning, A Fast and Accurate Dependency Parser using Neural Networks (Slides)

Content

1. Grammars and syntax parsing
2. Parsing with context free grammars
3. Parsing with probabilistic context free grammars
4. Dependency parsing
5. Parsing with neural networks