Dependency Parsing as Head Selection

Xingxing Zhang, Jianpeng Cheng, Mirella Lapata

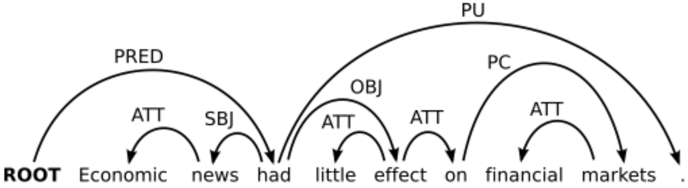
Institute for Language, Cognition and Computation
University of Edinburgh

x.zhang@ed.ac.uk

April 6, 2017

Dependency Parsing

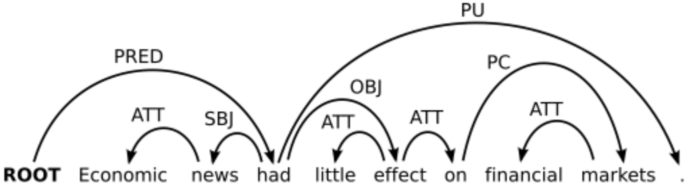
Dependency Parsing is the task of transforming a sentence $S = (\text{ROOT}, w_1, w_2, \ldots, w_N)$ into a directed tree originating out of ROOT.



- Parsing Algorithms
  - Transition-based Parsing
  - Graph-based Parsing

Dependency Parsing

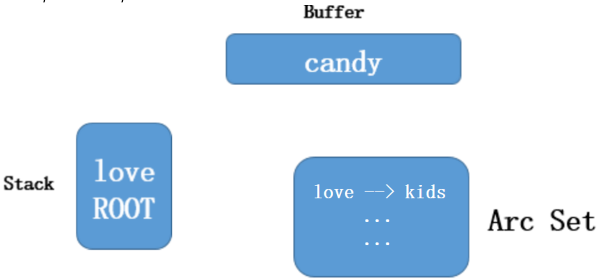
Dependency Parsing is the task of transforming a sentence $S = (\text{ROOT}, w_1, w_2, \ldots, w_N)$ into a directed tree originating out of ROOT.



- Parsing Algorithms
  - Transition-based Parsing
  - Graph-based Parsing
- Our parser is neither Transition-based nor Graph-based (during training)

Transition-based Parsing

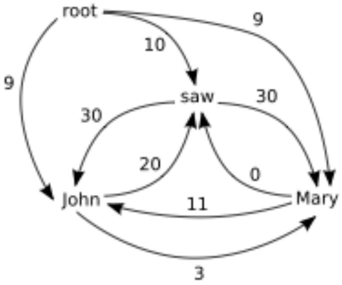
- Data Structure
  - Buffer, Stack, Arc Set



- Parsing:
  - Choose an action from
    SHIFT
    REDUCE-Left
    REDUCE-Right

Graph-based Parsing

- A Sentence → A Directed Complete Graph



(Graphs from Kubler et al., 2009)

- Parsing: Finding Maximum Spanning Tree
  - Chu-Liu-Edmond algorithm (Chu and Liu, 1965)
  - Eisner algorithm (Eisner 1996)

Recent Advances

Mostly replacing discrete features with Neural Network features.

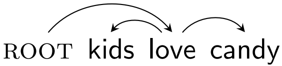
- Transition-based Parsers
  - Feed-Forward NN features (Chen and Manning, 2014)
  - Bi-LSTM features (Kiperwasser and Goldberg, 2016)
  - Stack LSTM: Buffer, Stack and Action Sequences modeled by Stack-LSTMs (Dyer et al., 2015)

- Graph-based Parsers
  - Tensor Decomposition features (Lei et al., 2014)
  - Feed-Forward NN features (Pei et al., 2015)
  - Bi-LSTM features (Kiperwasser and Goldberg, 2016)

Do we need a transition system or graph algorithm?

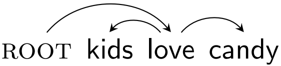
ROOT kids love candy

Do we need a transition system or graph algorithm?



- An important fact: Every word has only one head!

Do we need a transition system or graph algorithm?



- An important fact: Every word has only one head!
- Why not just learn to select the head?

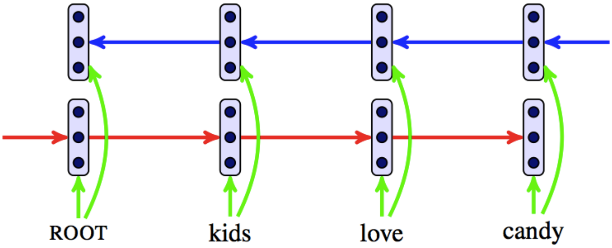
Dependency Parsing as Head Selection

DENSE: **De**pendency **N**eural **Se**lection

ROOT kids love candy

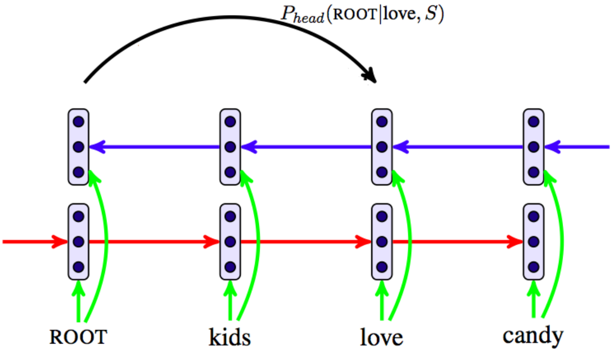
Dependency Parsing as Head Selection

DENSE: **De**pendency **N**eural **Se**lection



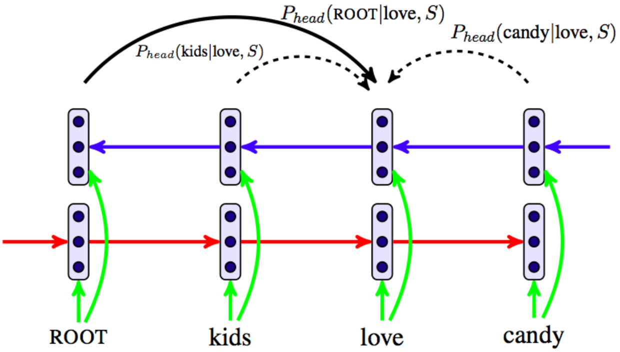
Dependency Parsing as Head Selection

DENSE: **De**pendency **N**eural **Se**lection



Dependency Parsing as Head Selection

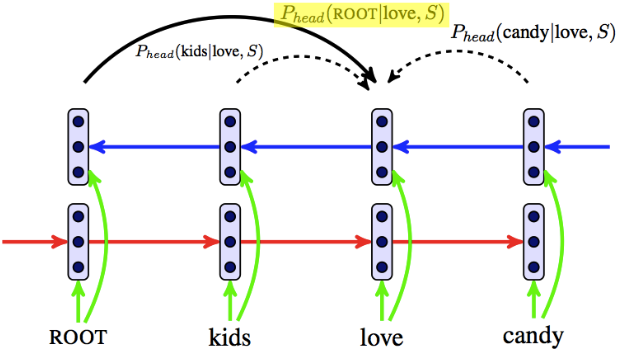
DENSE: **De**pendency **N**eural **Se**lection



$$P_{head}(\mathrm{ROOT}|\mathrm{love}, S) = \frac{\exp(MLP(\mathbf{a}_{\mathrm{ROOT}}, \mathbf{a}_{\mathrm{love}}))}{\sum_{k=0}^{3} \exp(MLP(\mathbf{a}_k, \mathbf{a}_{\mathrm{love}}))}$$

Dependency Parsing as Head Selection

DeNSe: **De**pendency **N**eural **Se**lection



$$P_{head}(\text{ROOT}|\text{love}, S) = \frac{\exp(MLP(\mathbf{a}_{\text{ROOT}}, \mathbf{a}_{\text{love}}))}{\sum_{k=0}^{3} \exp(MLP(\mathbf{a}_k, \mathbf{a}_{\text{love}}))}$$

Decoding

- Greedy Decoding: The output may not be a (projective) tree!

Decoding

- Greedy Decoding: The output may not be a (projective) tree!

Dataset	#Sent (Dev)	Greedy Decoding Tree	Proj
PTB (English)	1,700	95.1	<u>86.6</u>
CTB (Chinese)	803	87.0	<u>73.1</u>
Czech	374	<u>87.7</u>	65.5
German	367	<u>96.7</u>	67.3

Decoding

- Greedy Decoding: The output may not be a (projective) tree!

Dataset	#Sent (Dev)	Greedy Decoding Tree	Greedy Decoding Proj
PTB (English)	1,700	95.1	<u>86.6</u>
CTB (Chinese)	803	87.0	<u>73.1</u>
Czech	374	<u>87.7</u>	65.5
German	367	<u>96.7</u>	67.3

- Decoding with a Maximum Spanning Tree Algorithm (relatively rare)
  - Projective Parsing: Eisner Algorithm
  - Non-projective Parsing: Chu-Liu-Edmond Algorithm

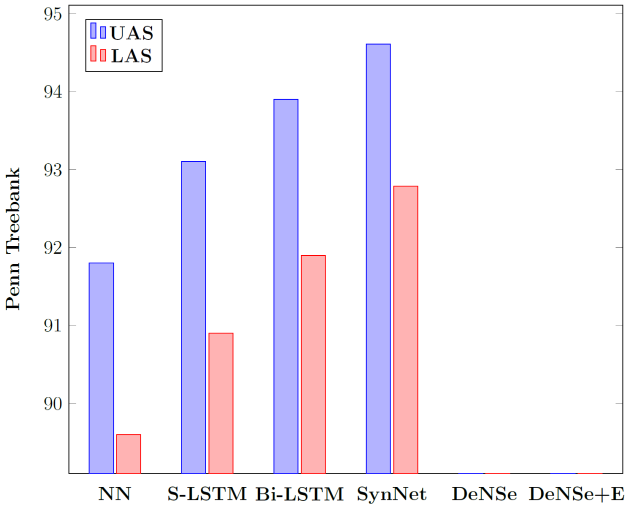
Labelled Parser

A two-layer Rectifier Network (Glorot et al., 2011)

- Dependent Word:
  - Bi-LSTM Feature
  - Word Embedding
  - PoS Embedding
- Head Word:
  - Bi-LSTM Feature
  - Word Embedding
  - PoS Embedding

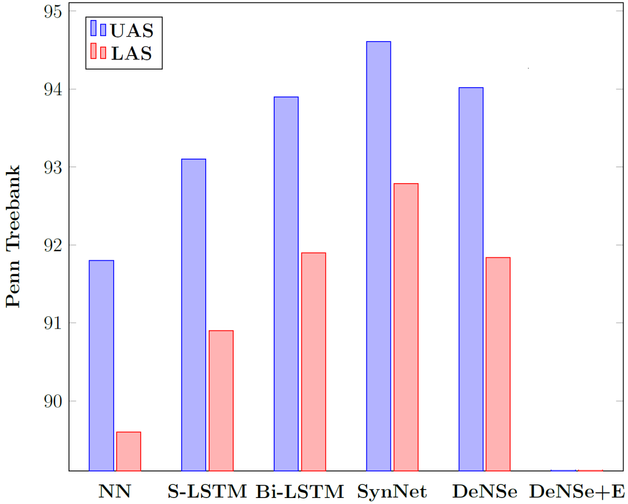
Experiments

Projective Parsing Results (PTB; English)



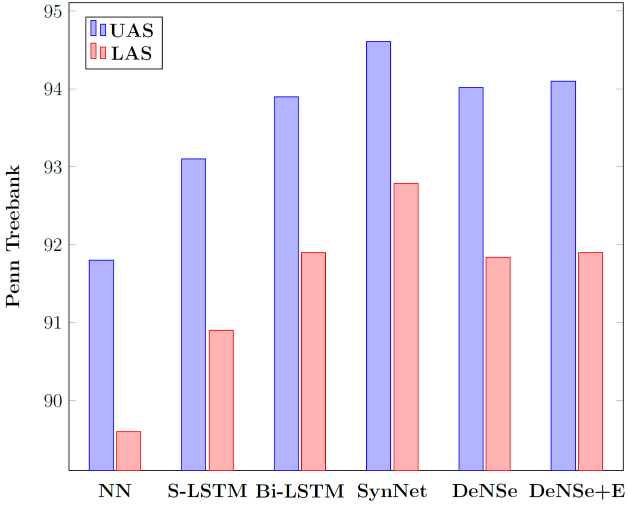
NN (Chen & Manning, 2014); S-LSTM (Dyer et al., 2015);
Bi-LSTM (Kiperwasser & Goldberg, 2016); SynNet (Andor et al. 2016)

Projective Parsing Results (PTB; English)



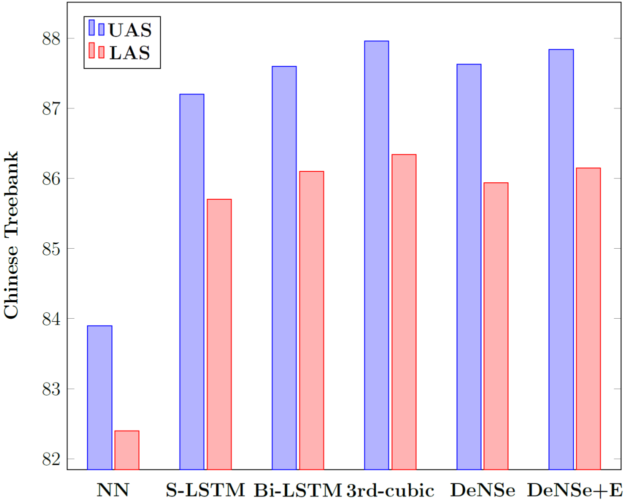
NN (Chen & Manning, 2014); S-LSTM (Dyer et al., 2015);
Bi-LSTM (Kiperwasser & Goldberg, 2016); SynNet (Andor et al. 2016)

Projective Parsing Results (PTB; English)



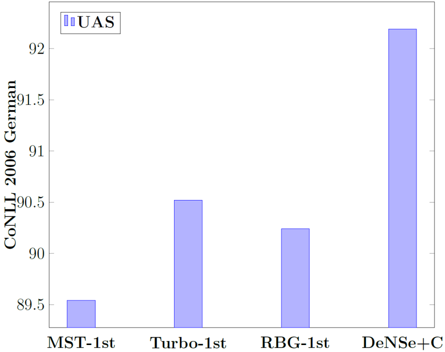
NN (Chen & Manning, 2014); S-LSTM (Dyer et al., 2015);
Bi-LSTM (Kiperwasser & Goldberg, 2016); SynNet (Andor et al. 2016)

Projective Parsing Results (PTB; Chinese)



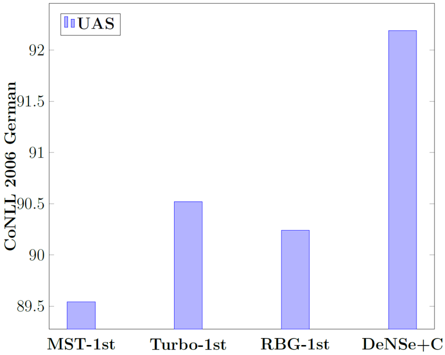
NN (Chen & Manning, 2014); S-LSTM (Dyer et al., 2015); Bi-LSTM (Kiperwasser & Goldberg, 2016); 3rd-cubic (Zhang & McDonald 2014)

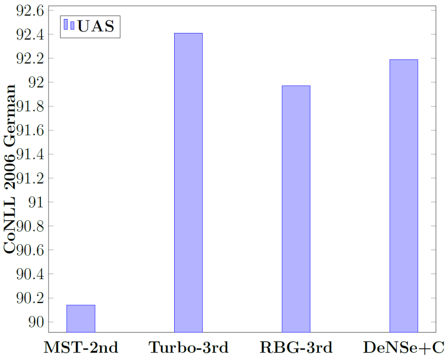
Non-projective Parsing Results (German)



MST-1st, MST-2nd (McDonald et al., 2005) Turbo-1st, Turbo-3rd (Martins et al., 2013) RBG-1st RBG-3rd (Martins et al. 2013)

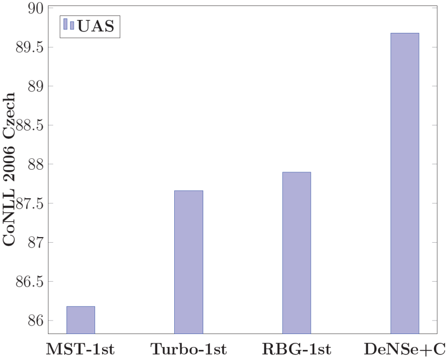
Non-projective Parsing Results (German)





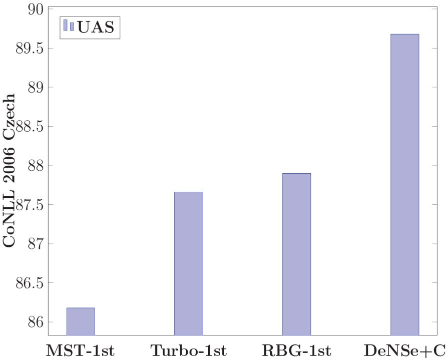
MST-1st, MST-2nd (McDonald et al., 2005) Turbo-1st, Turbo-3rd (Martins et al., 2013) RBG-1st RBG-3rd (Martins et al. 2013)

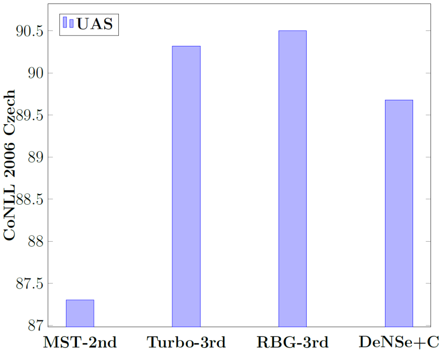
Non-projective Parsing Results (Czech)



MST-1st, MST-2nd ((McDonald et al., 2005) Turbo-1st, Turbo-3rd (Martins et al., 2013) RBG-1st RBG-3rd (Martins et al. 2013)

Non-projective Parsing Results (Czech)





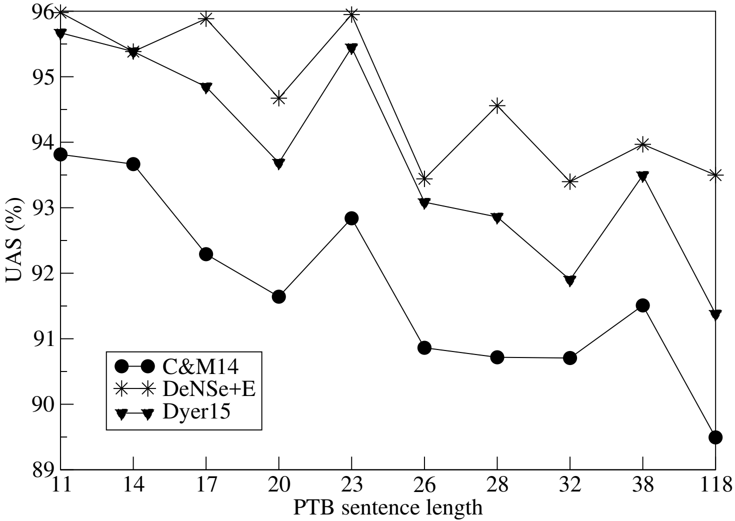
MST-1st, MST-2nd ((McDonald et al., 2005) Turbo-1st, Turbo-3rd (Martins et al., 2013) RBG-1st RBG-3rd (Martins et al. 2013)

Unlabeled Exact Match

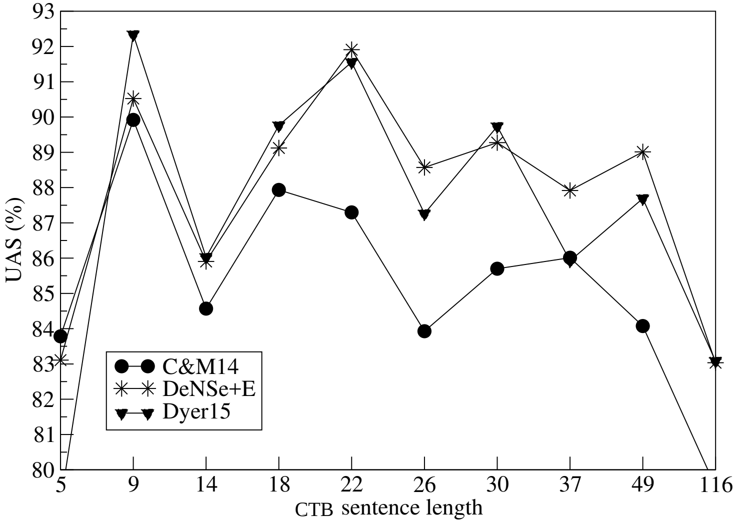
Parser	PTB		CTB	
	Dev	Test	Dev	Test
C&M14	43.35	40.93	32.75	32.20
Dyer15	51.94	50.70	**39.72**	**37.23**
DeNSe	51.24	49.34	34.74	33.66
DeNSe+E	**52.47**	**50.79**	36.49	35.13

Table: UEM results on PTB and CTB.

UAS v.s. Length



UAS v.s. Length



Conclusions

- We propose a dependency parser as greedily selecting the head of each word in sentence.
- Combine the greedy model with a MST algorithm can further increase the performance
- Code available: https://github.com/XingxingZhang/dense_parser

Thanks

Q & A