Top-down Tree Long Short-Term Memory Networks

Xingxing Zhang, Liang Lu, Mirella Lapata

School of Informatics, University of Edinburgh

12th June, 2016

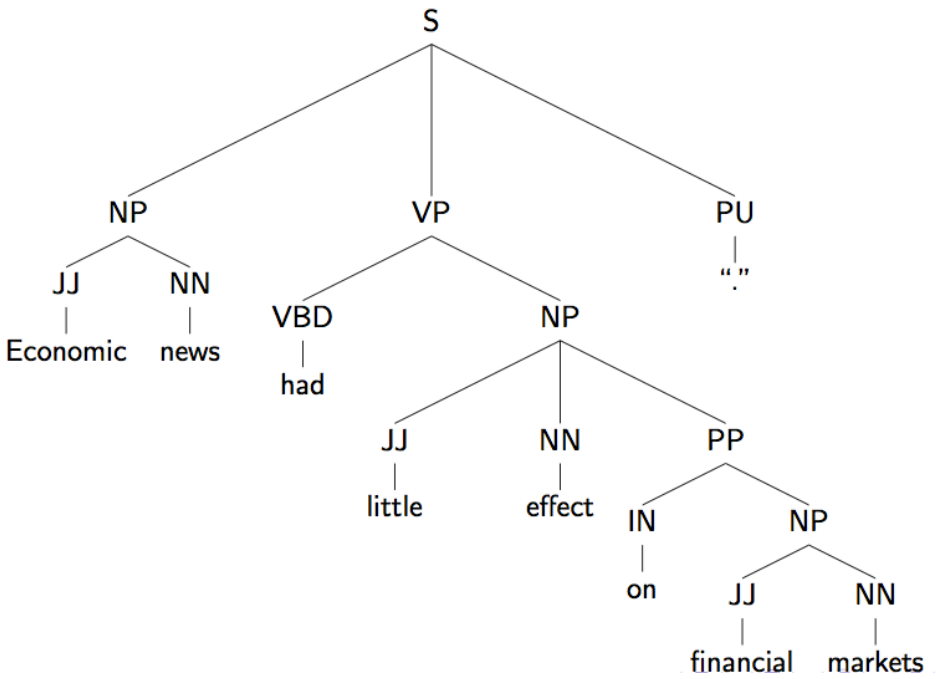
Sequential Language Models

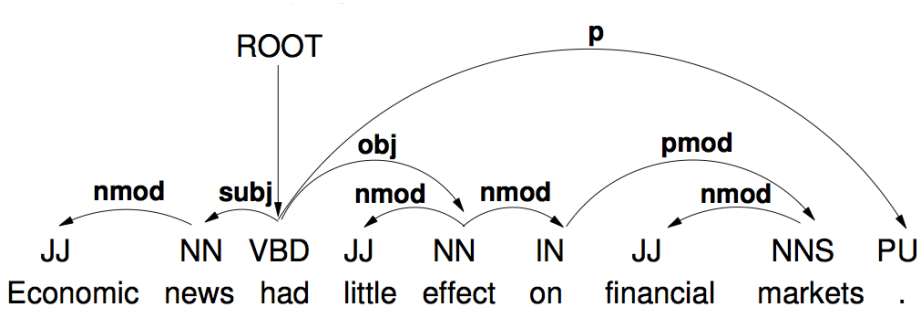
$$P(S = w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} P(w_i | w_{1:i-1}) \tag{1}$$

- State of the Art
  - based on Long Short Term Memory Network Language Model (Hochreiter and Schmidhuber, 1997; Sundermeyer et al., 2012)
  - **Billion word benchmark results** reported in Jozefowicz et al., (2016)

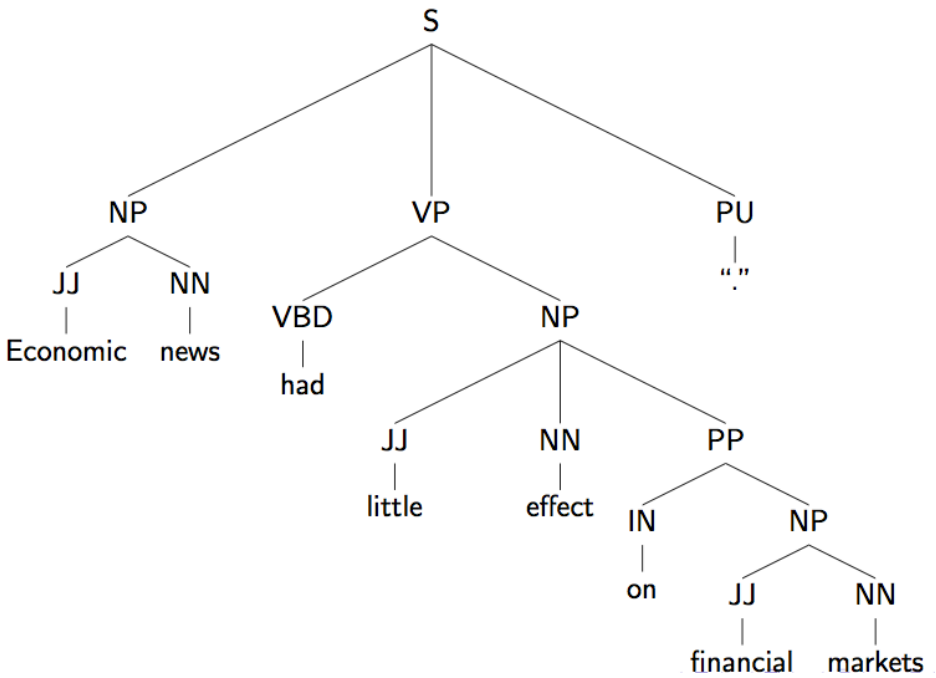
Models	PPL
KN5	67.6
LSTM	30.6
LSTM+CNN INPUTS	30.0

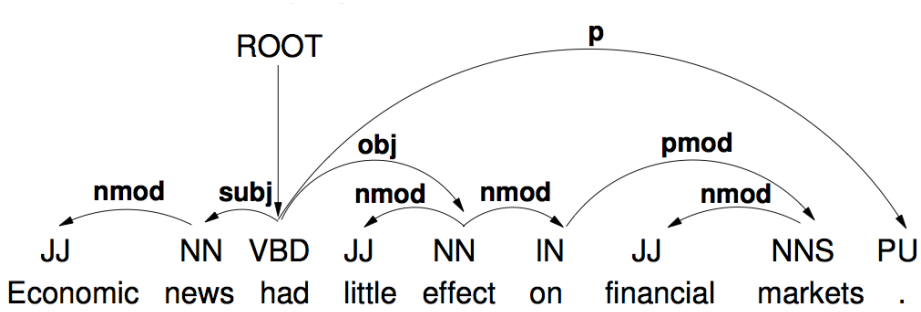
Will tree structures help LMs?





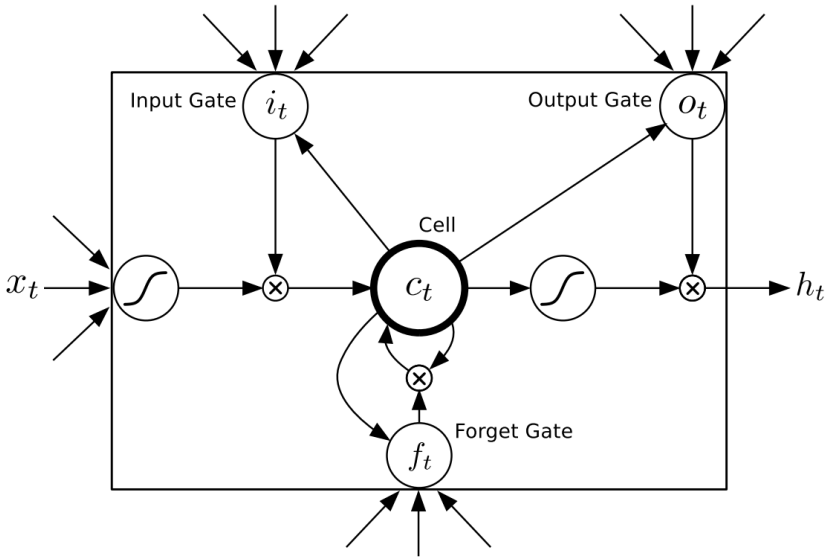
Will tree structures help LMs?

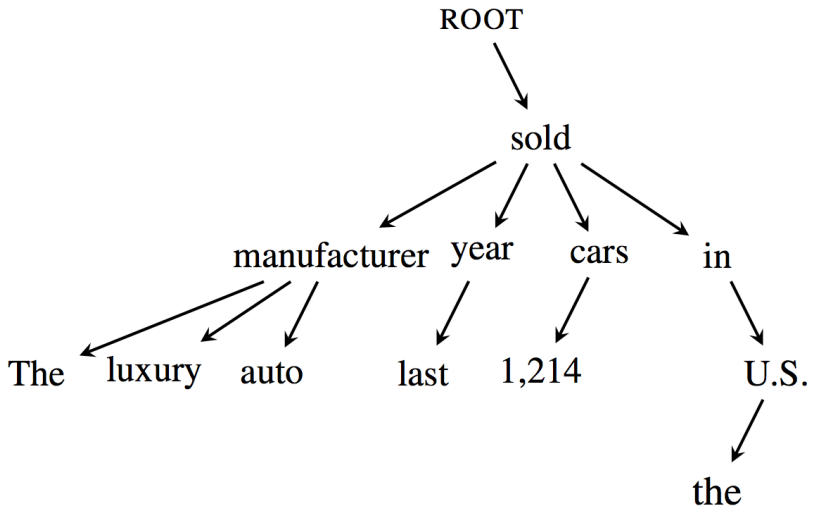




- Probably yes
  - LMs based on Constituency Parsing (Chelba and Jelinek, 2000; Roark, 2001; Charniak, 2001)
  - LMs based on Dependency Parsing (Shen et al., 2008; Zhang, 2009; Sennrich, 2015)

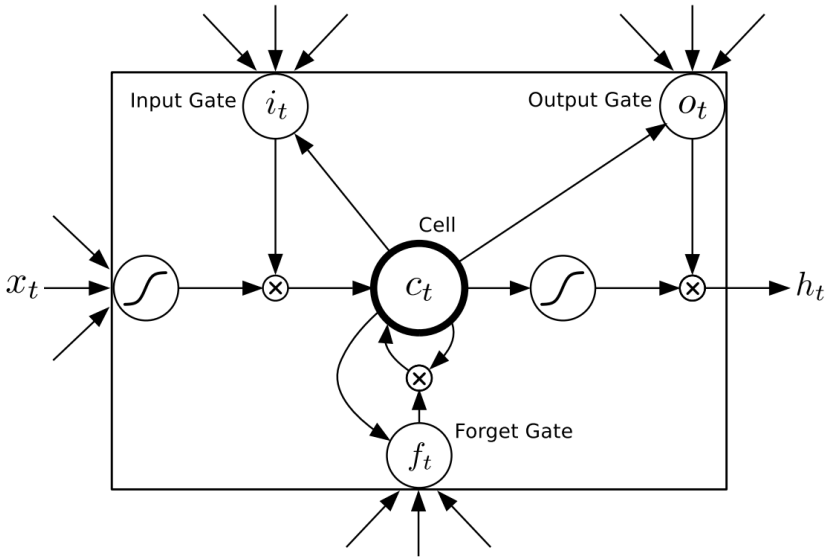
LSTMs + Dependency Trees = TreeLSTMs

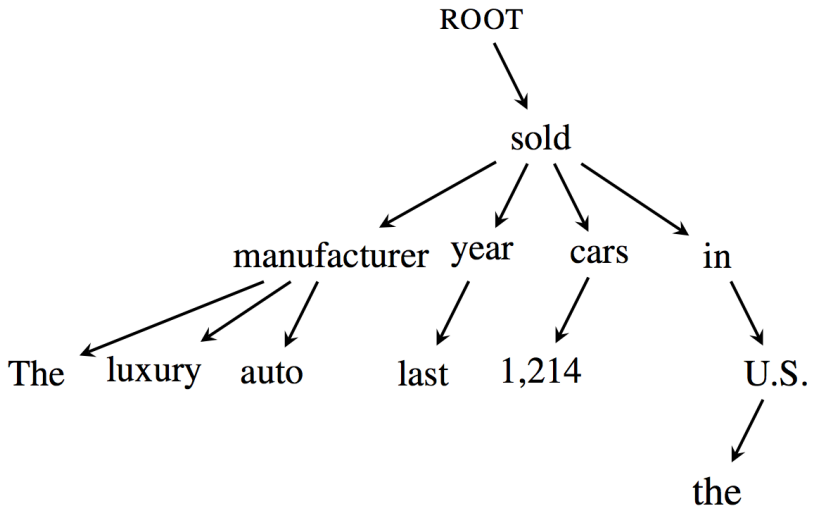




- Why?
  - Sentence Length N v.s. Tree Height $log(N)$

LSTMs + Dependency Trees = TreeLSTMs





- Why?
  - Sentence Length N v.s. Tree Height $log(N)$
- How?
  - Top-down Generation
  - Breadth-first search
  - reminiscent of Eisner (1996)

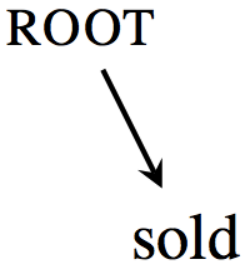
Generation Process (Unlabeled Trees)

The luxury auto manufacturer last year sold 1,214 cars in the U.S.

ROOT

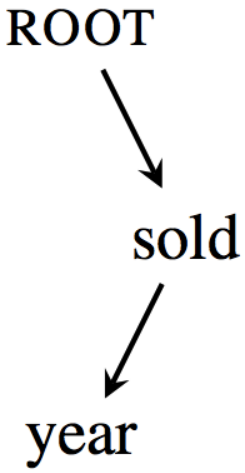
Generation Process (Unlabeled Trees)

The luxury auto manufacturer last year sold 1,214 cars in the U.S.



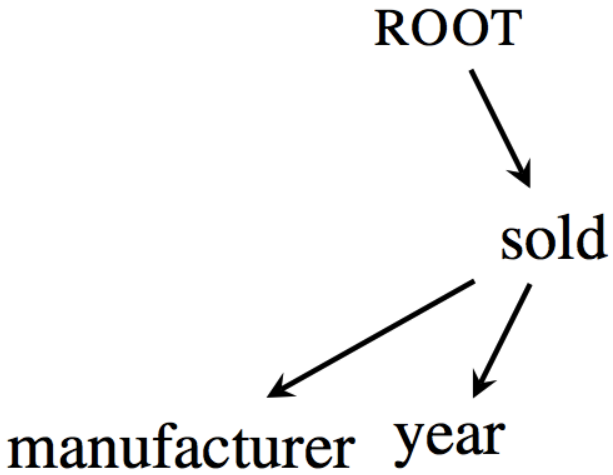
Generation Process (Unlabeled Trees)

The luxury auto manufacturer last year sold 1,214 cars in the U.S.



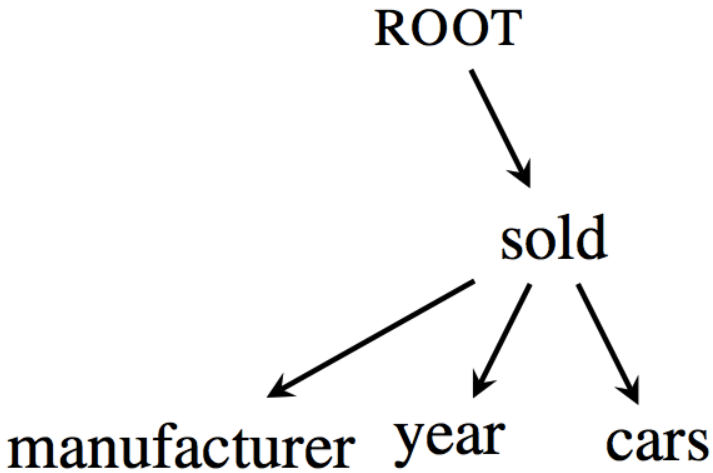
Generation Process (Unlabeled Trees)

The luxury auto manufacturer last year sold 1,214 cars in the U.S.



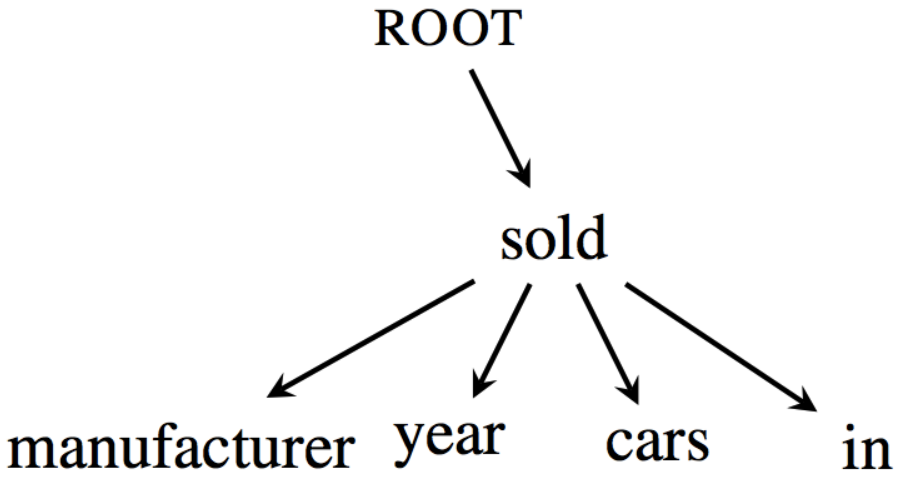
Generation Process (Unlabeled Trees)

The luxury auto manufacturer last year sold 1,214 cars in the U.S.



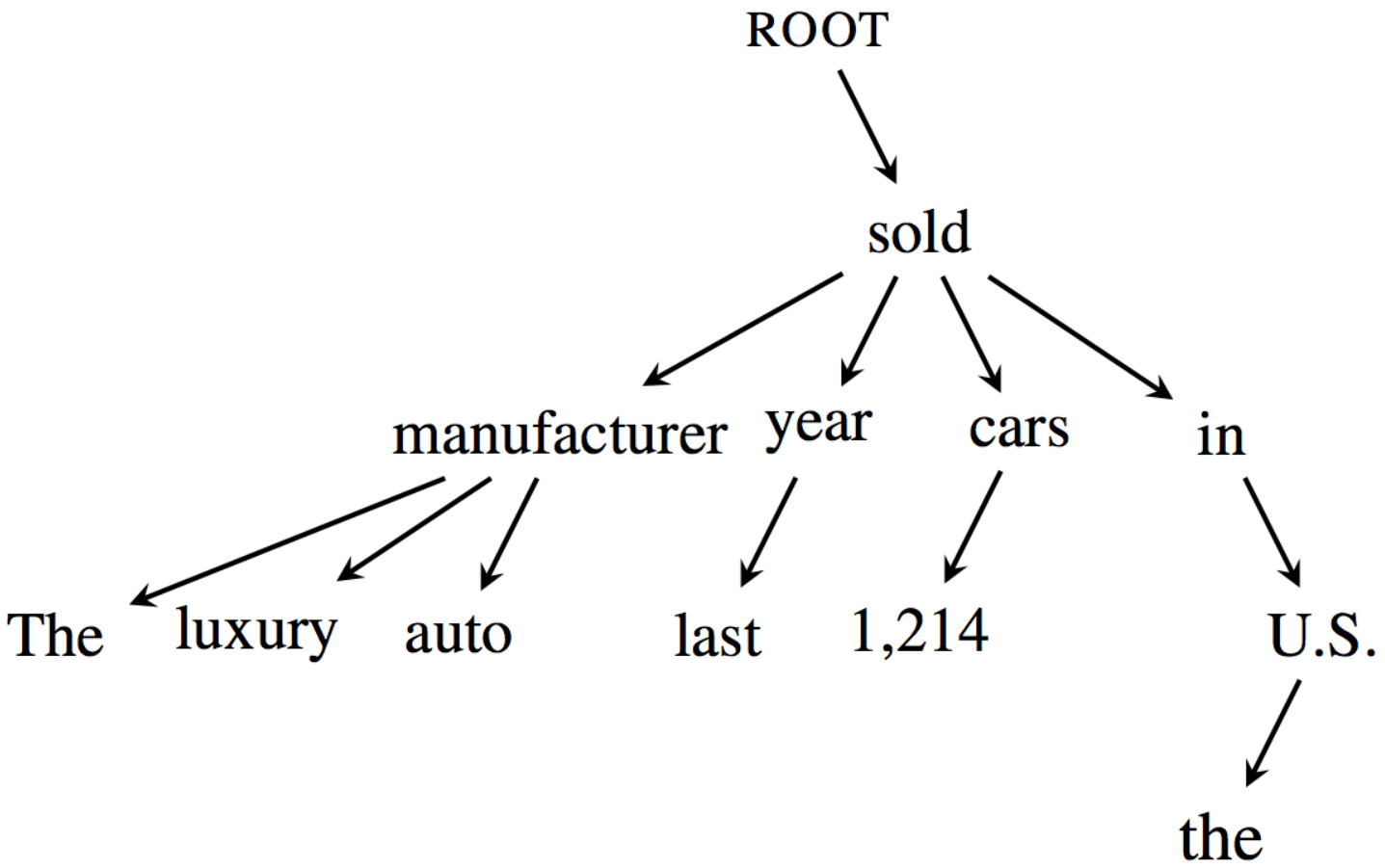
Generation Process (Unlabeled Trees)

The luxury auto manufacturer last year sold 1,214 cars in the U.S.



Generation Process (Unlabeled Trees)

The luxury auto manufacturer last year sold 1,214 cars in the U.S.



Tree LSTM

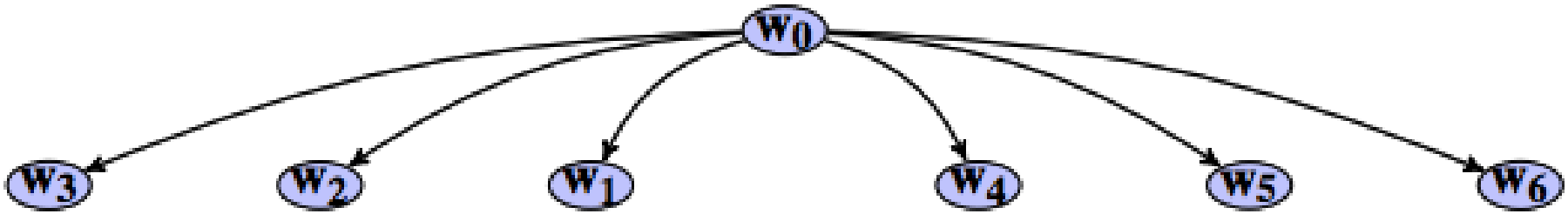
$$P(S) = \prod_{i=1}^{n} P(w_i | w_{1:i-1}) \tag{2}$$

$$\Downarrow$$

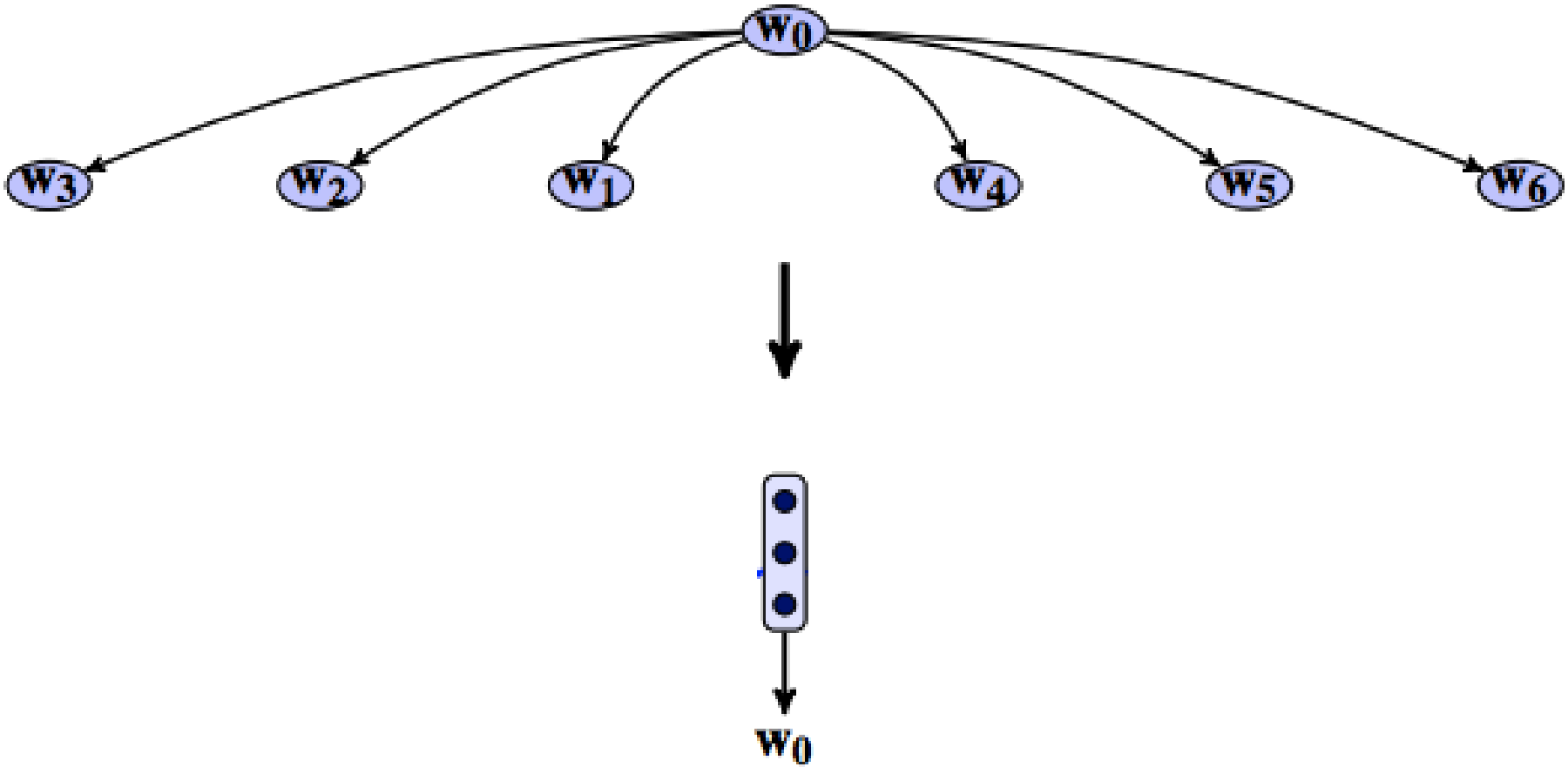
$$P(S|T) = \prod_{w \in \mathrm{BFS}(T) \setminus \mathrm{ROOT}} P(w|\mathcal{D}(w)) \tag{3}$$

- $\mathcal{D}(w)$ is the *Dependency Path* of w.
- $\mathcal{D}(w)$ is a generated sub-tree.
- Works on **projective** and **unlabeled** dependency trees.

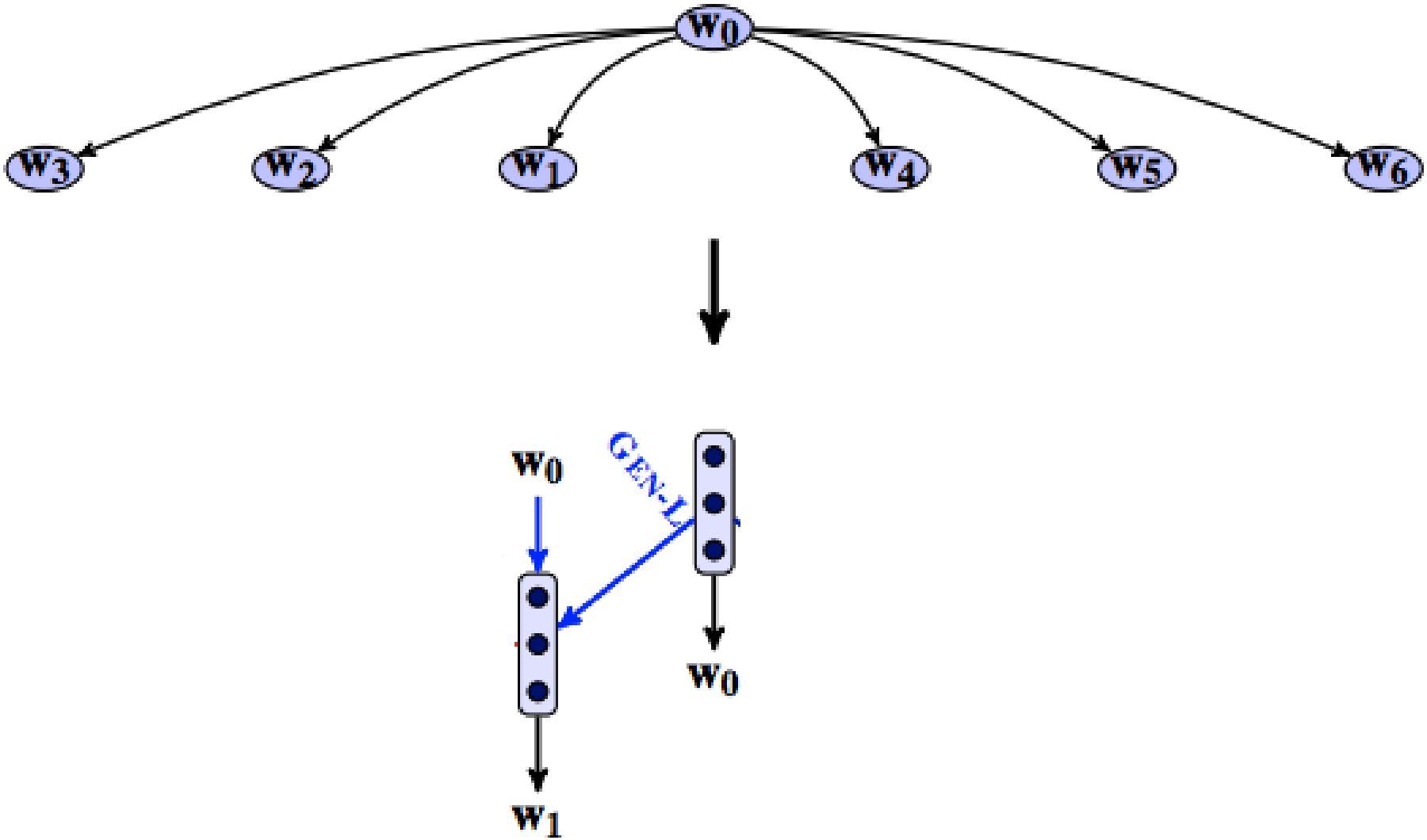
Tree LSTM



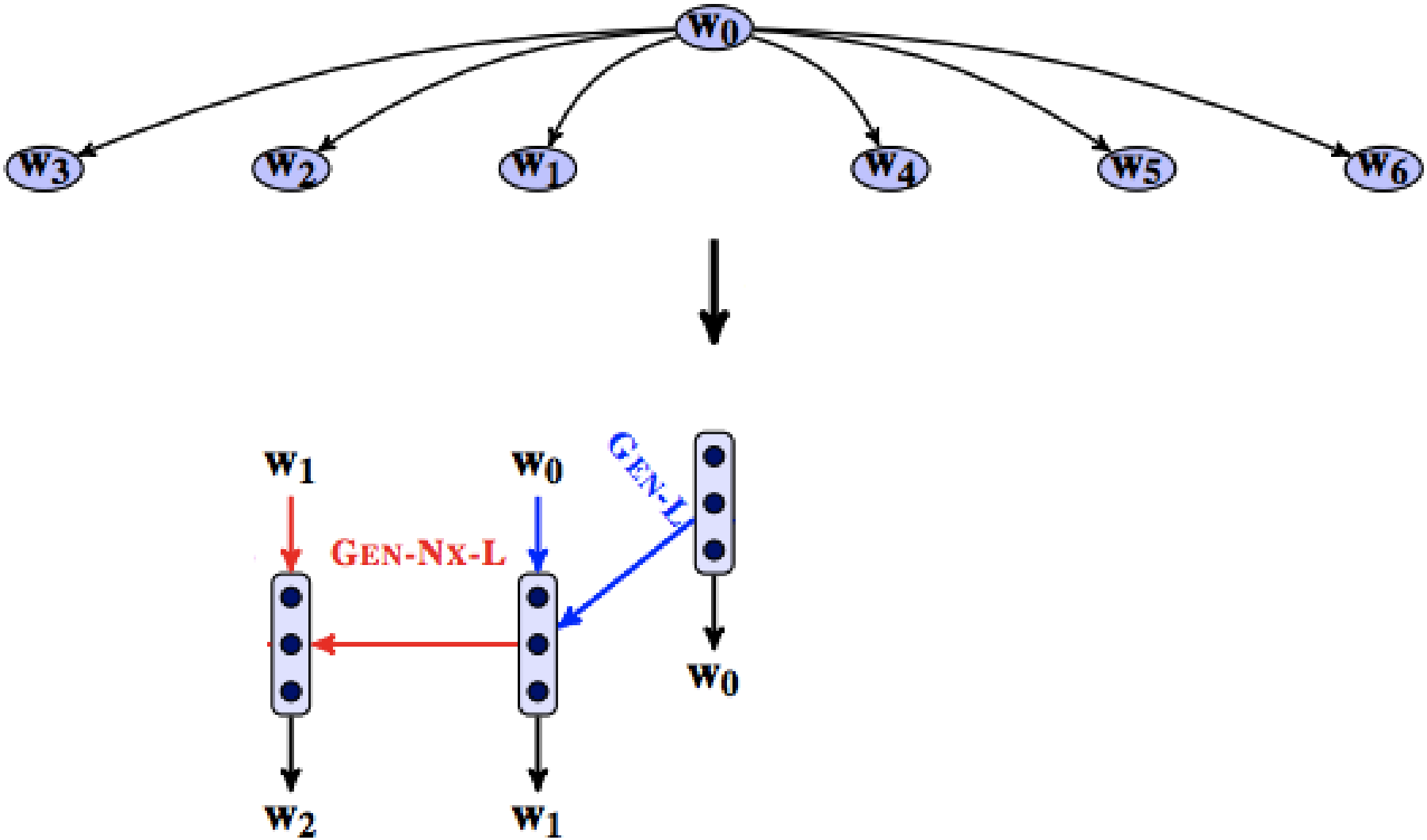
Tree LSTM



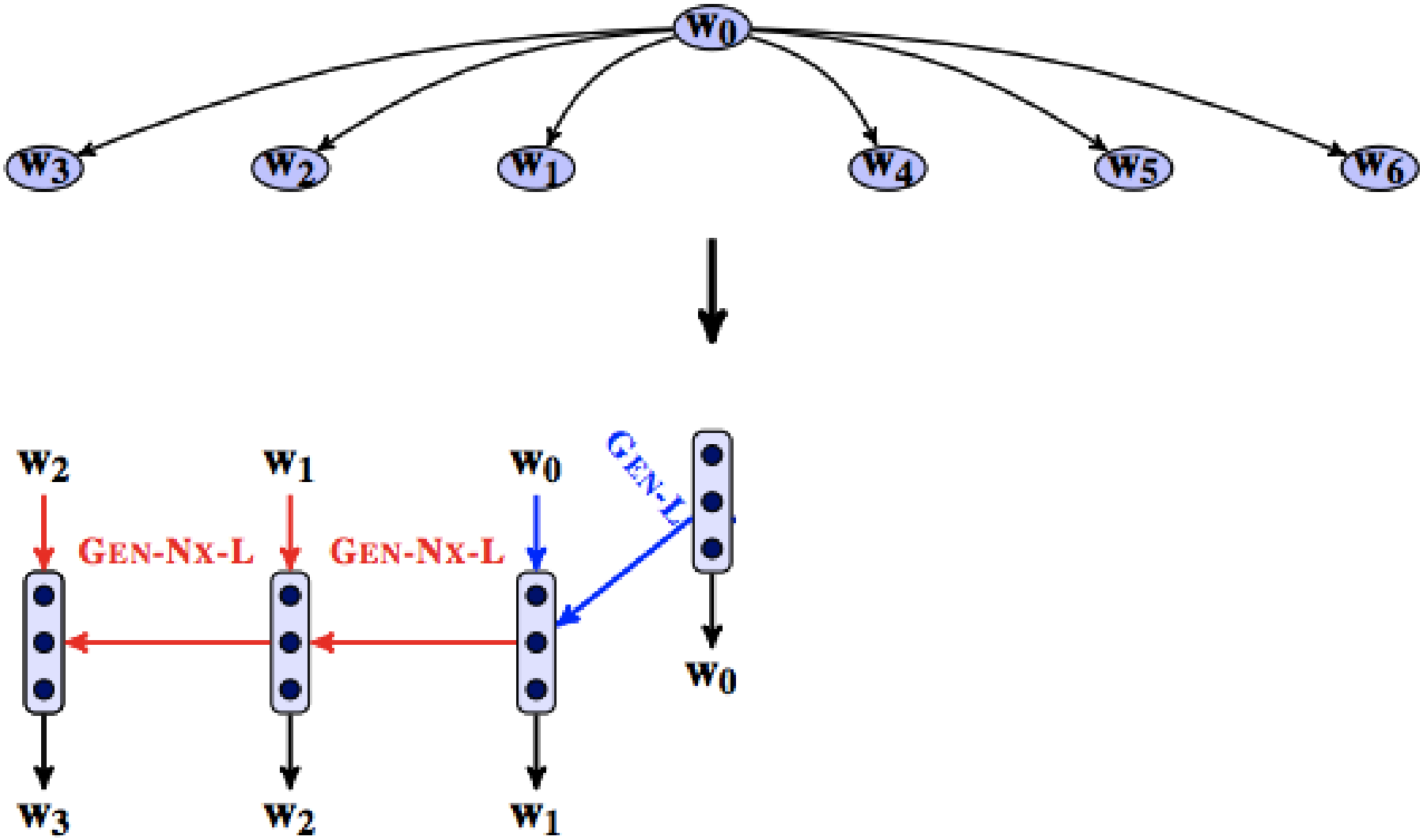
Tree LSTM



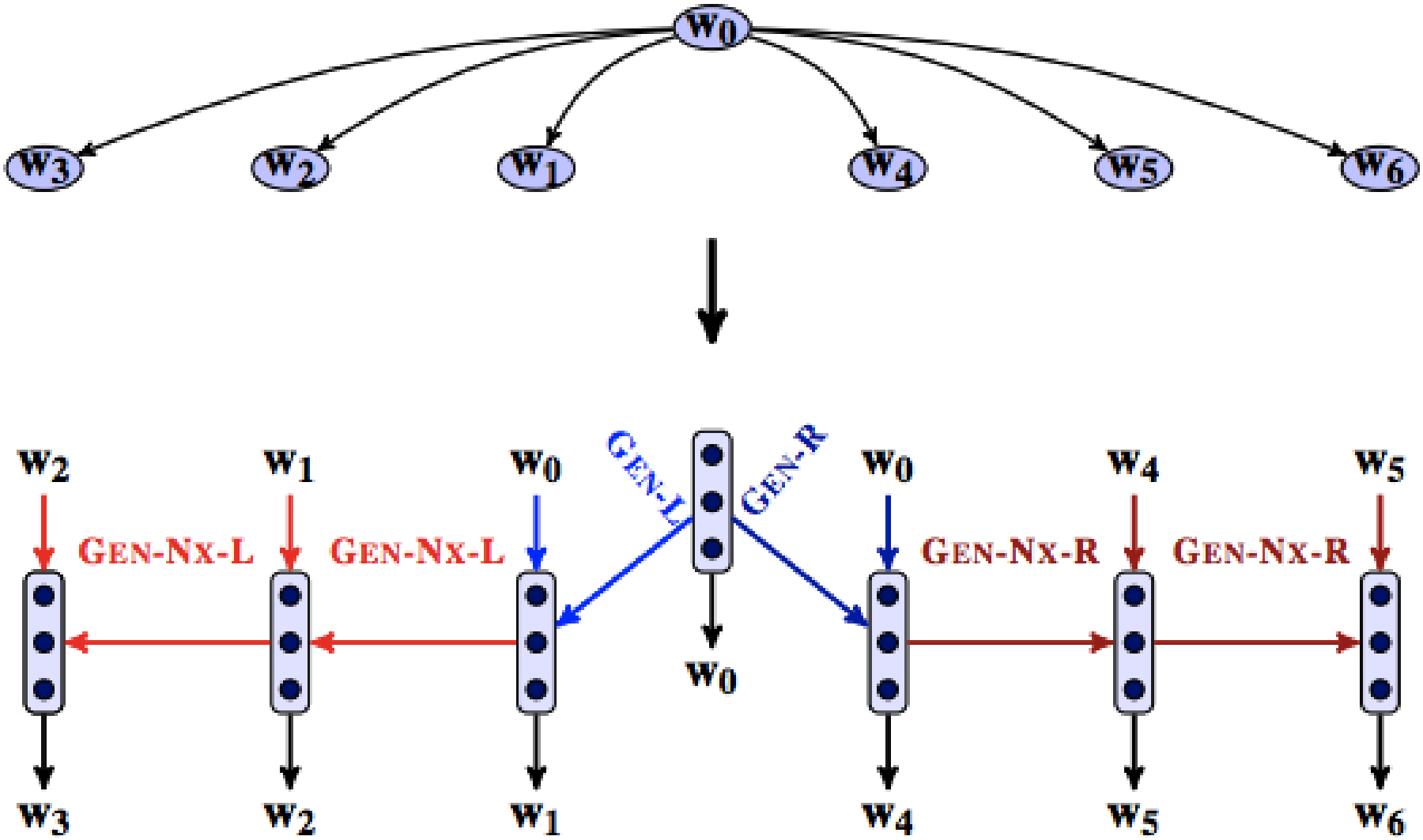
Tree LSTM



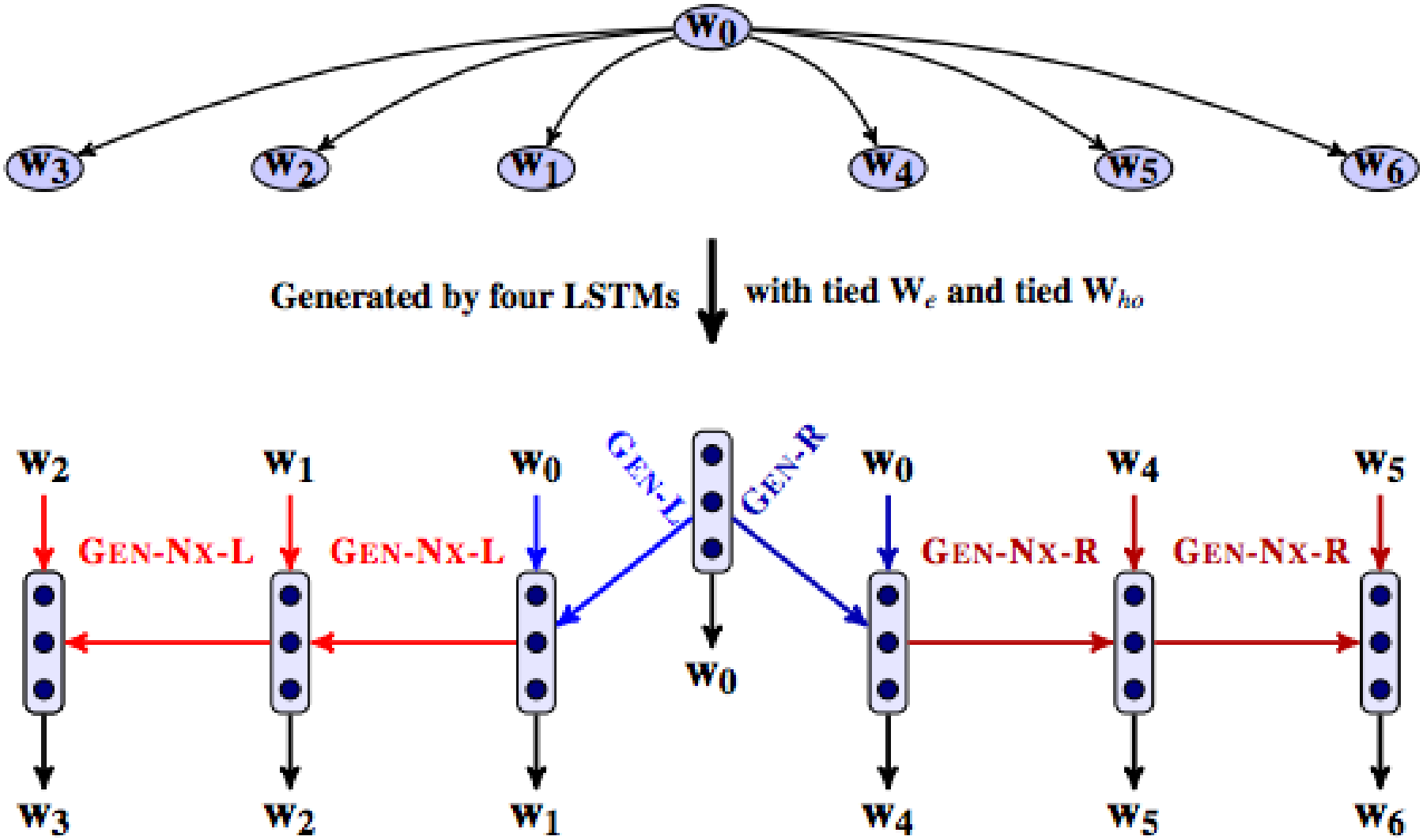
Tree LSTM



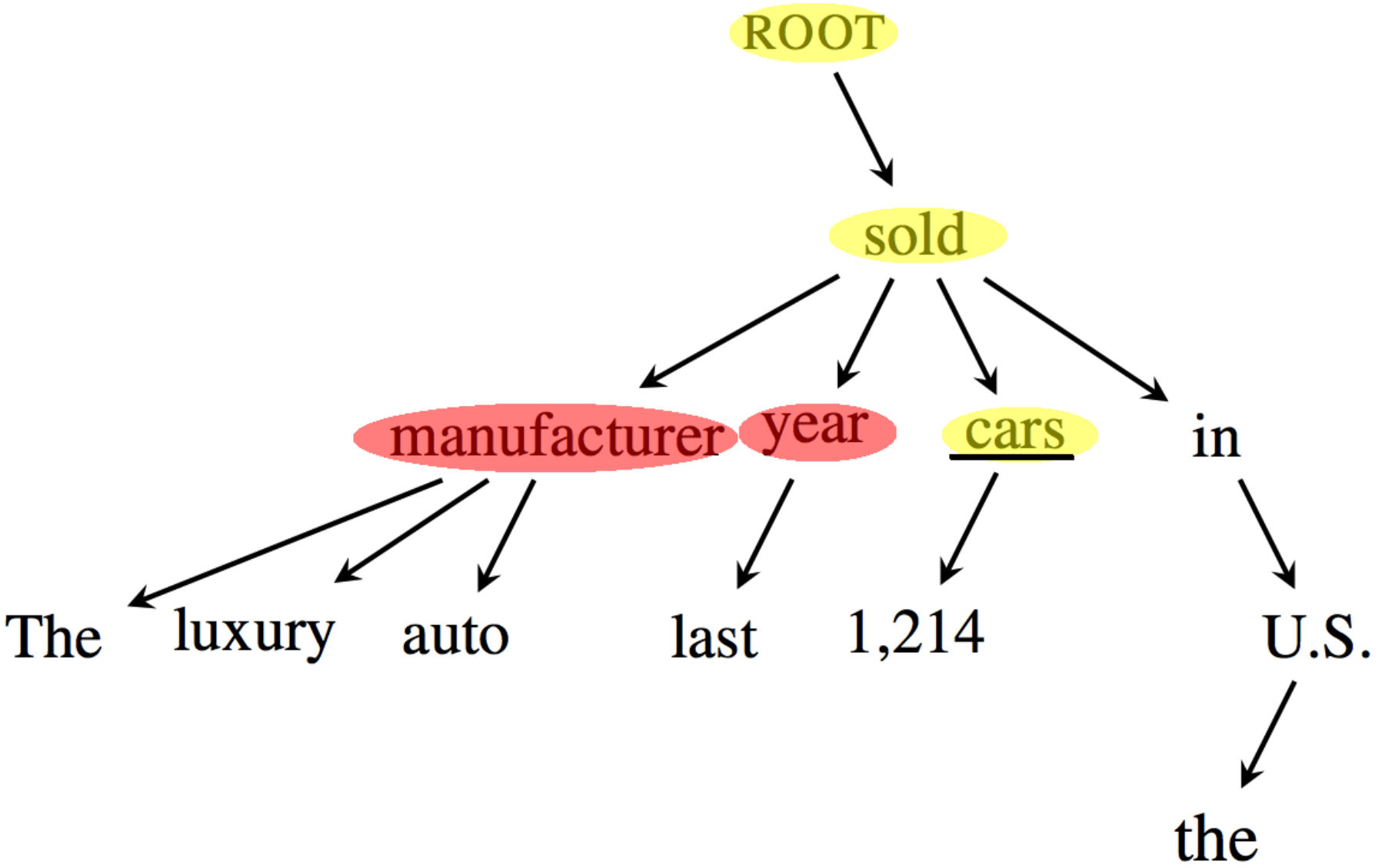
Tree LSTM


w0
w3
w2
w1
w4
w5
w6
w2
w1
w0
GEN-L
GEN-R
w0
w4
w5
GEN-NX-L
GEN-NX-L
GEN-NX-R
GEN-NX-R
w0
w3
w2
w1
w4
w5
w6

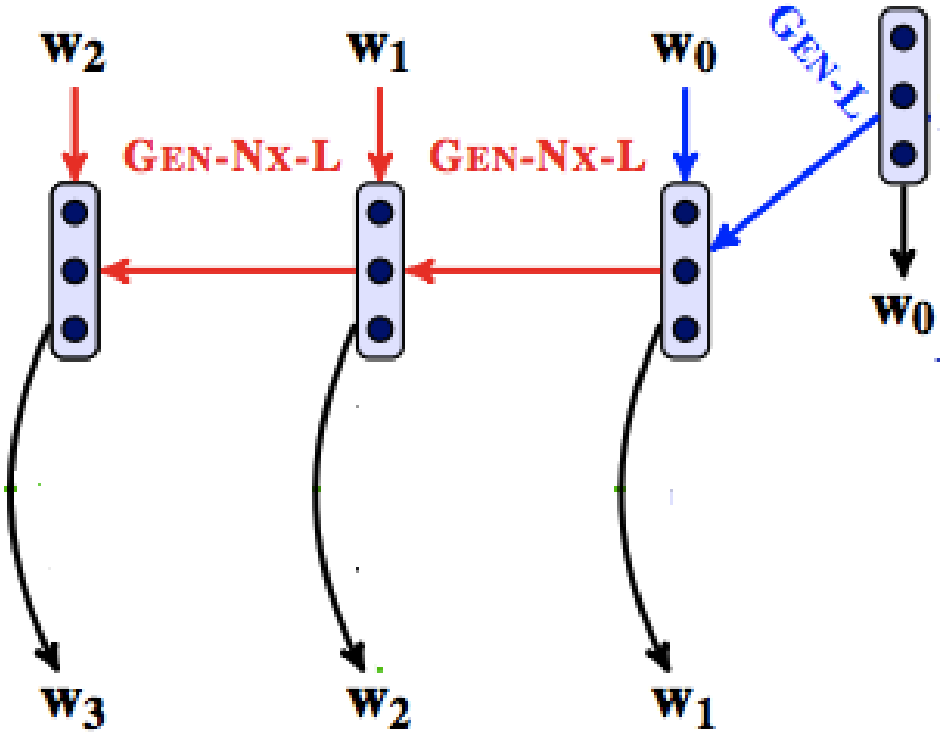
Tree LSTM


w0
w3
w2
w1
w4
w5
w6
Generated by four LSTMs
with tied W_e and tied W_{ho}
w2
w1
w0
Gen-L
Gen-R
w0
w4
w5
Gen-Nx-L
Gen-Nx-L
Gen-Nx-R
Gen-Nx-R
w0
w3
w2
w1
w4
w5
w6

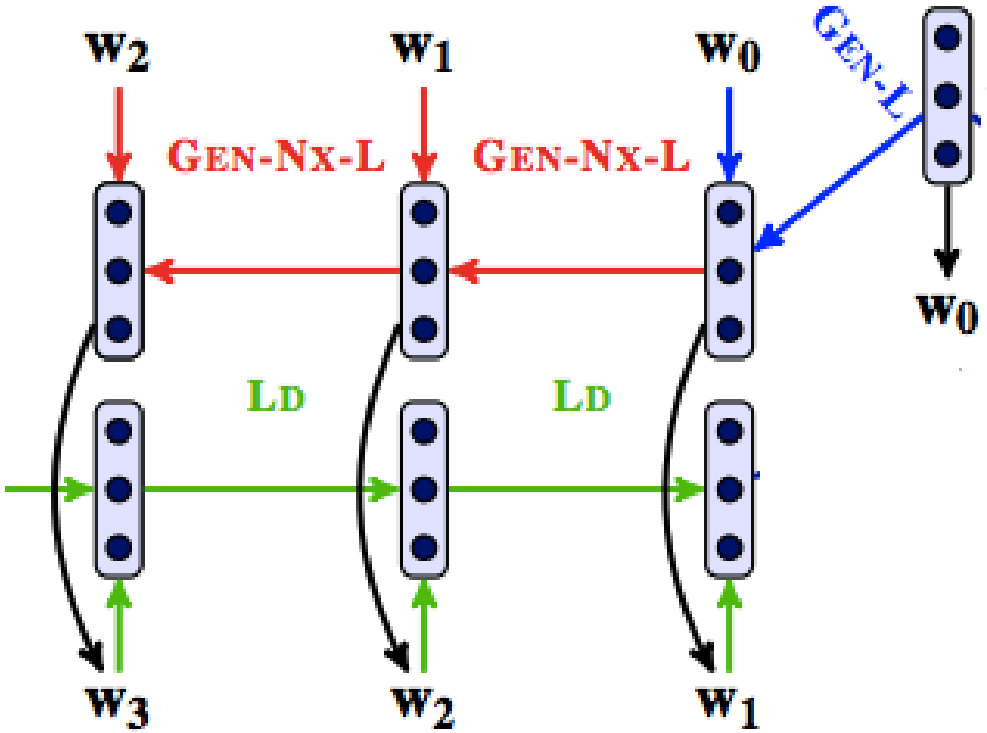
One Limitation of Tree LSTM



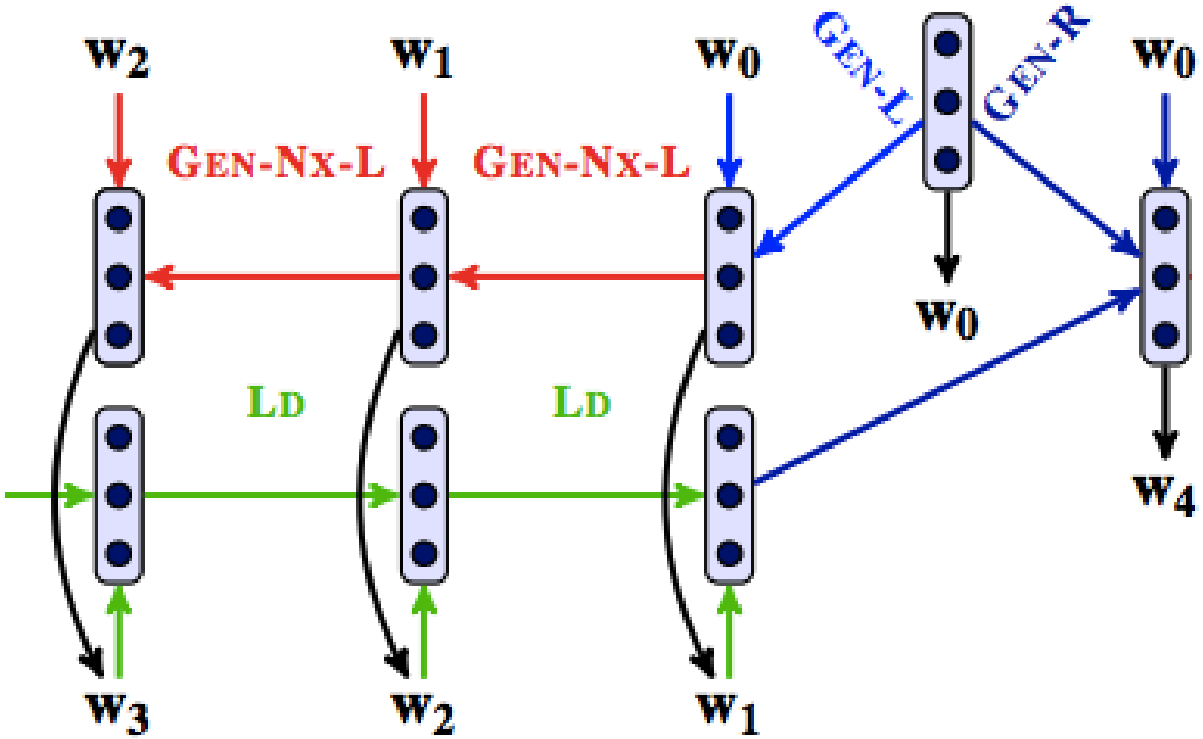
Left Dependent Tree LSTM



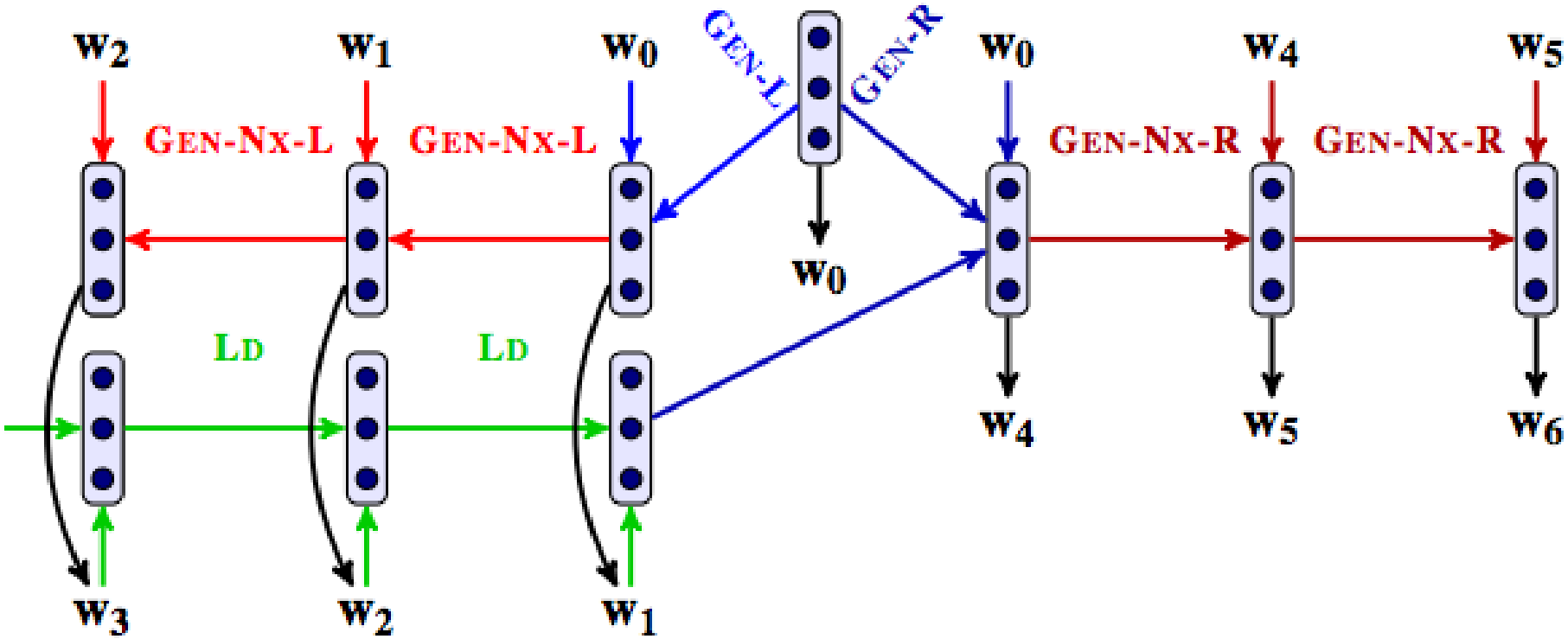
Left Dependent Tree LSTM



Left Dependent Tree LSTM



Left Dependent Tree LSTM

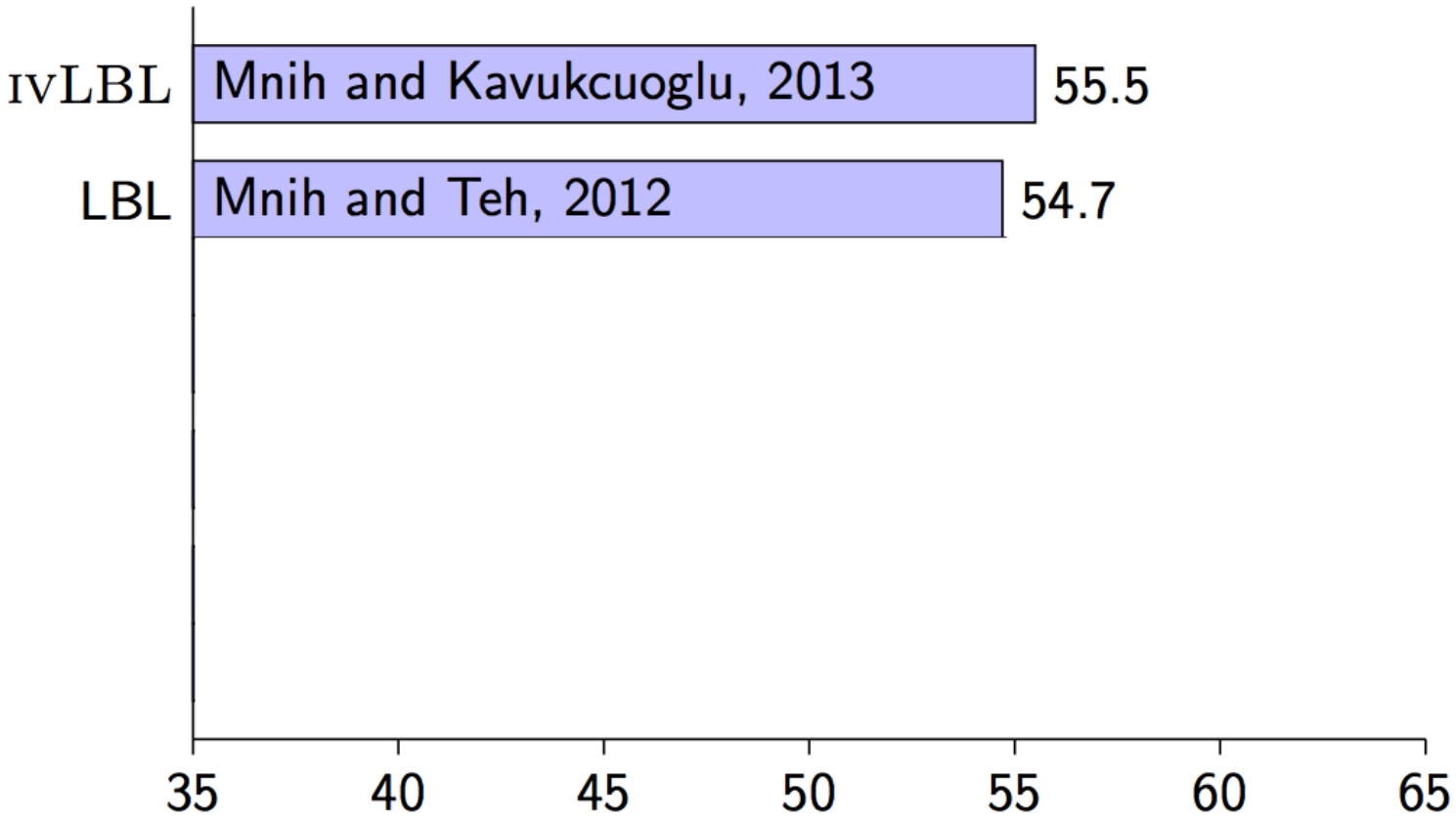


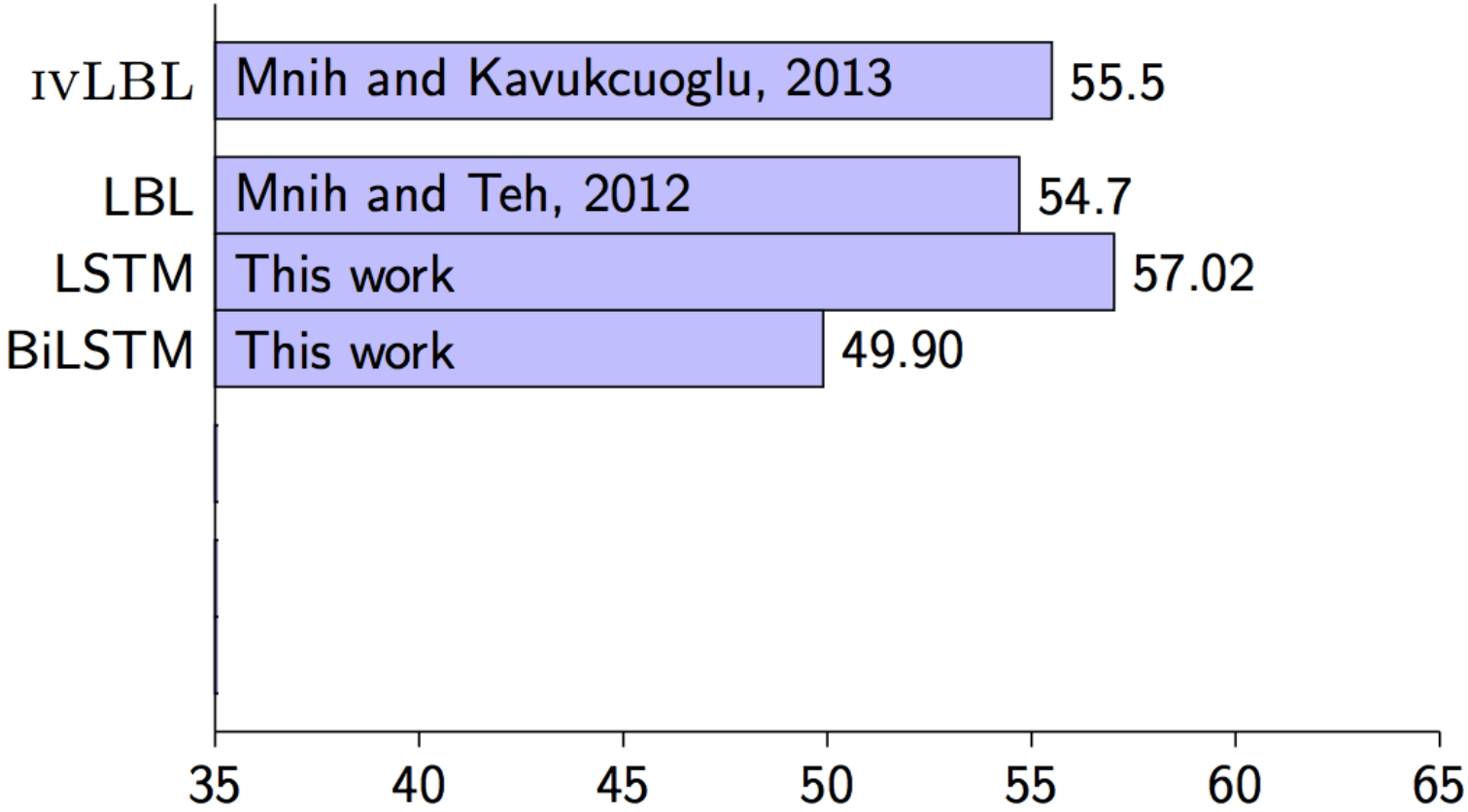
Experiments

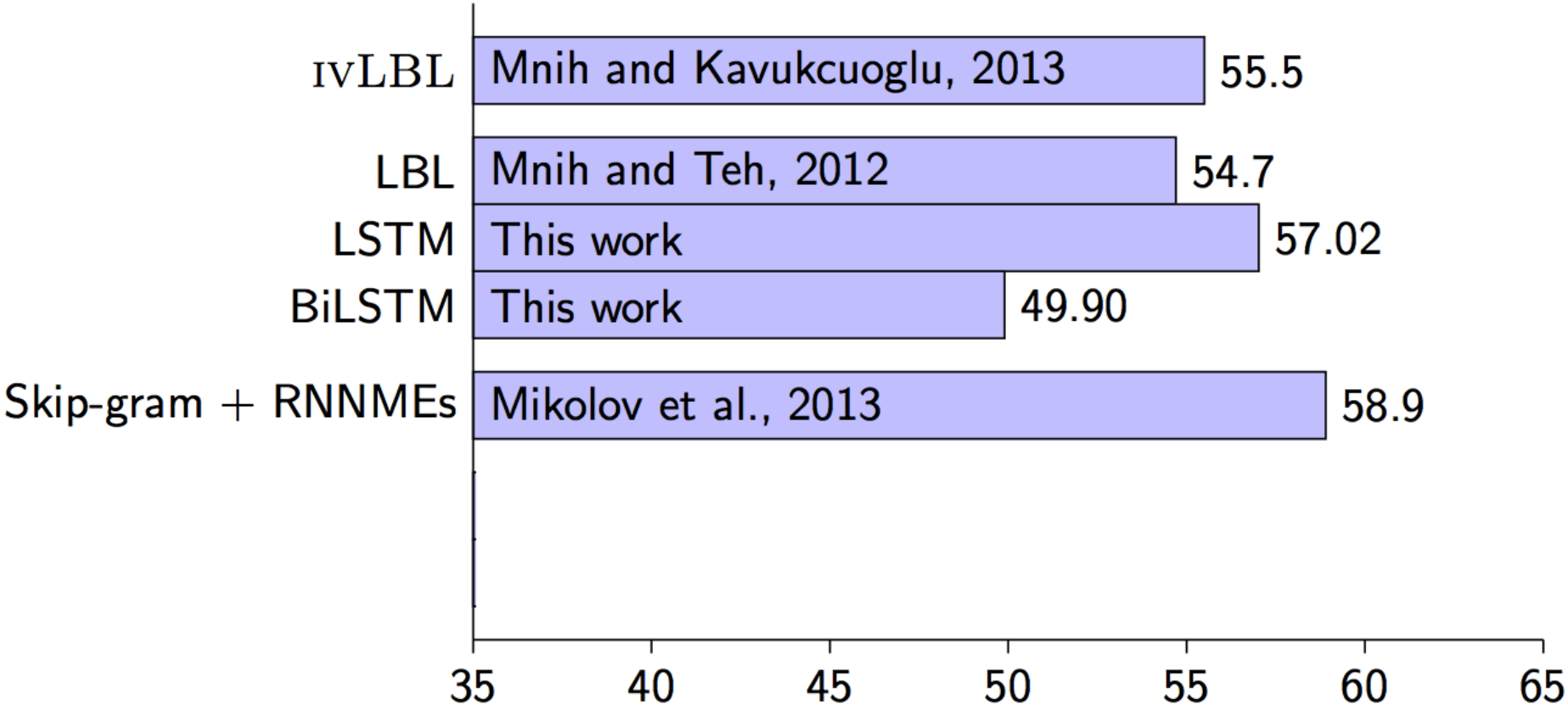
MSR Sentence Completion Challenge

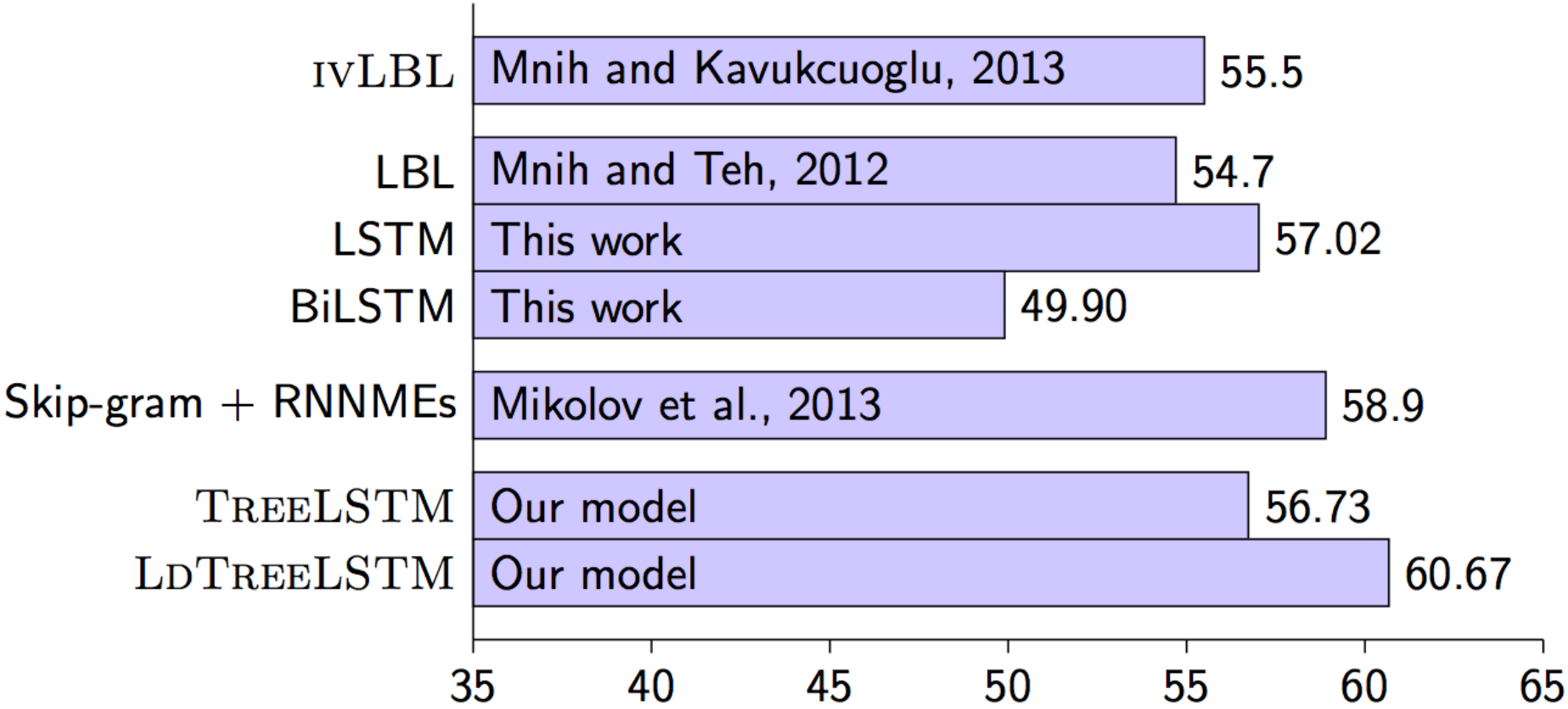
1) I have seen it on him , and could _____ to it.
a) write b) migrate c) climb d) swear e) contribute

- Training set: 49 million words (around 2 million sentences)
- development set: 4000 sentences
- test set: 1040 completion questions.


IVLBL
Mnih and Kavukcuoglu, 2013
55.5
LBL
Mnih and Teh, 2012
54.7
35
40
45
50
55
60
65


ivLBL
Mnih and Kavukcuoglu, 2013
55.5
LBL
Mnih and Teh, 2012
54.7
LSTM
This work
57.02
BiLSTM
This work
49.90
35
40
45
50
55
60
65

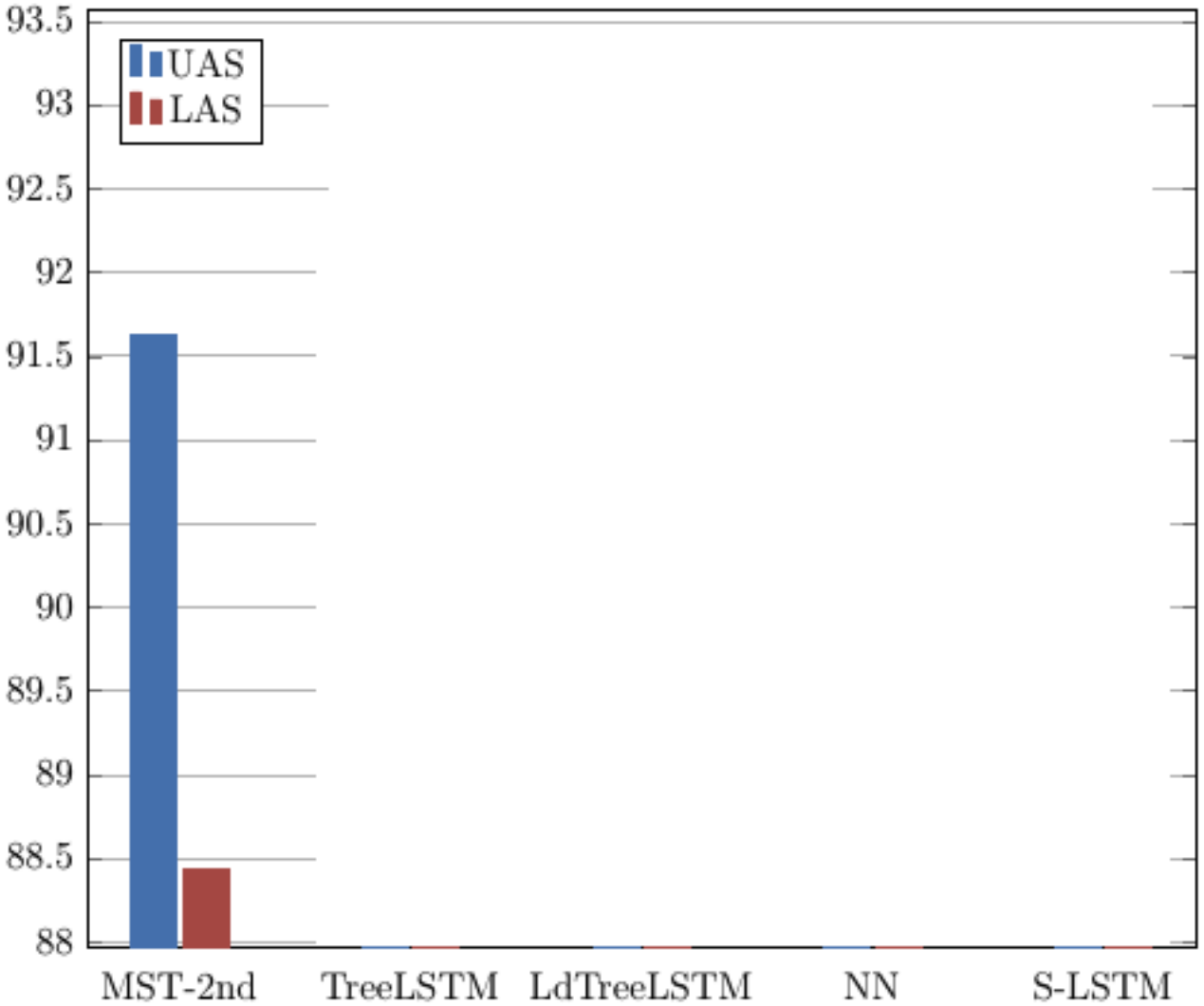

ivLBL
Mnih and Kavukcuoglu, 2013
55.5
LBL
Mnih and Teh, 2012
54.7
LSTM
This work
57.02
BiLSTM
This work
49.90
Skip-gram + RNNMEs
Mikolov et al., 2013
58.9
35
40
45
50
55
60
65


ivLBL
Mnih and Kavukcuoglu, 2013
55.5
LBL
Mnih and Teh, 2012
54.7
LSTM
This work
57.02
BiLSTM
This work
49.90
Skip-gram + RNNMEs
Mikolov et al., 2013
58.9
TreeLSTM
Our model
56.73
LdTreeLSTM
Our model
60.67
35
40
45
50
55
60
65

Dependency Parsing Reranking

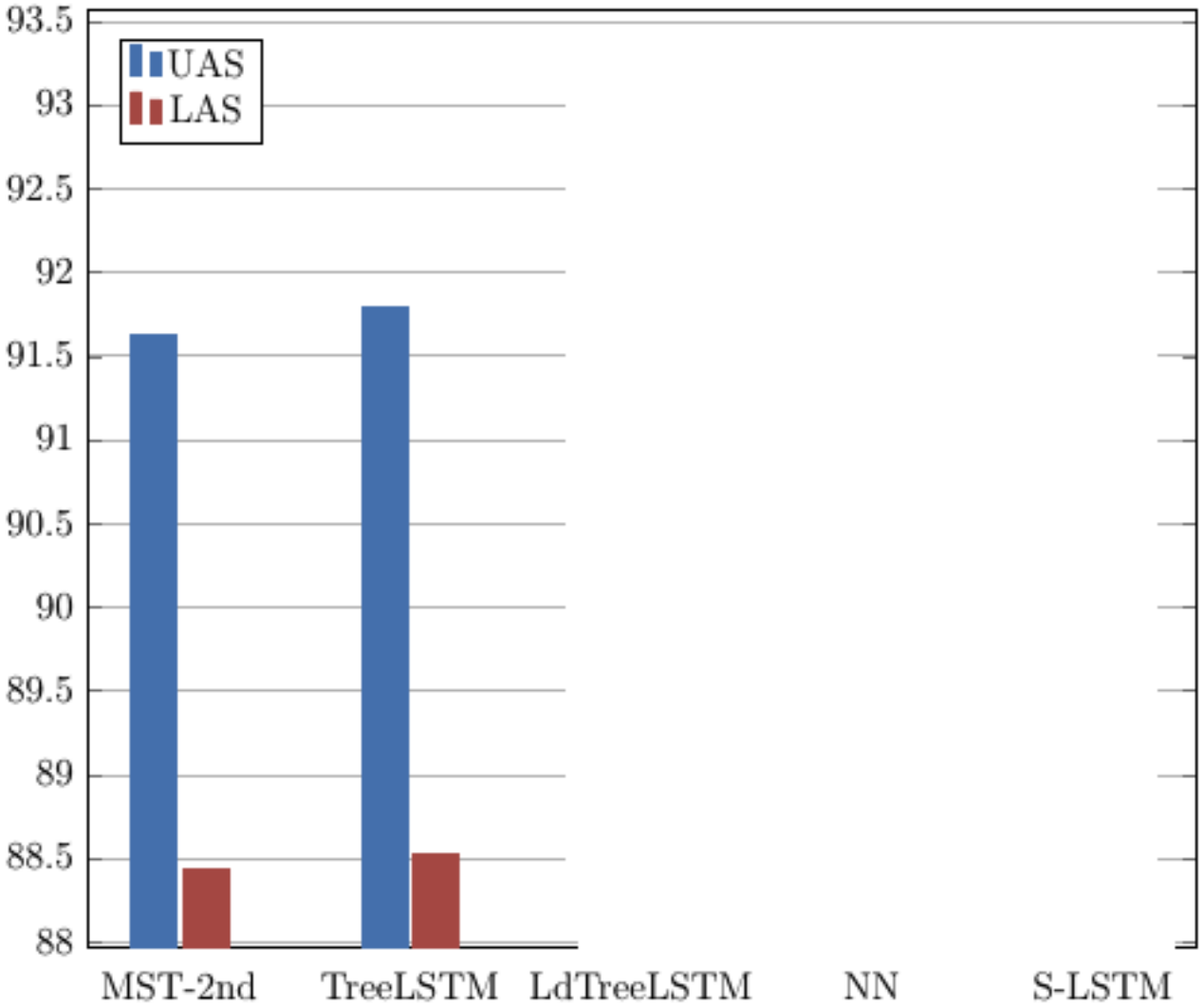
- Rerank 2nd Order MSTParser (McDonald and Pereira, 2006)
- We train TreeLSTM and LdTreeLSTM as language models.
- We only use words as input features; POS tags, dependency labels or composition features are not used.

Dependency Parsing Reranking



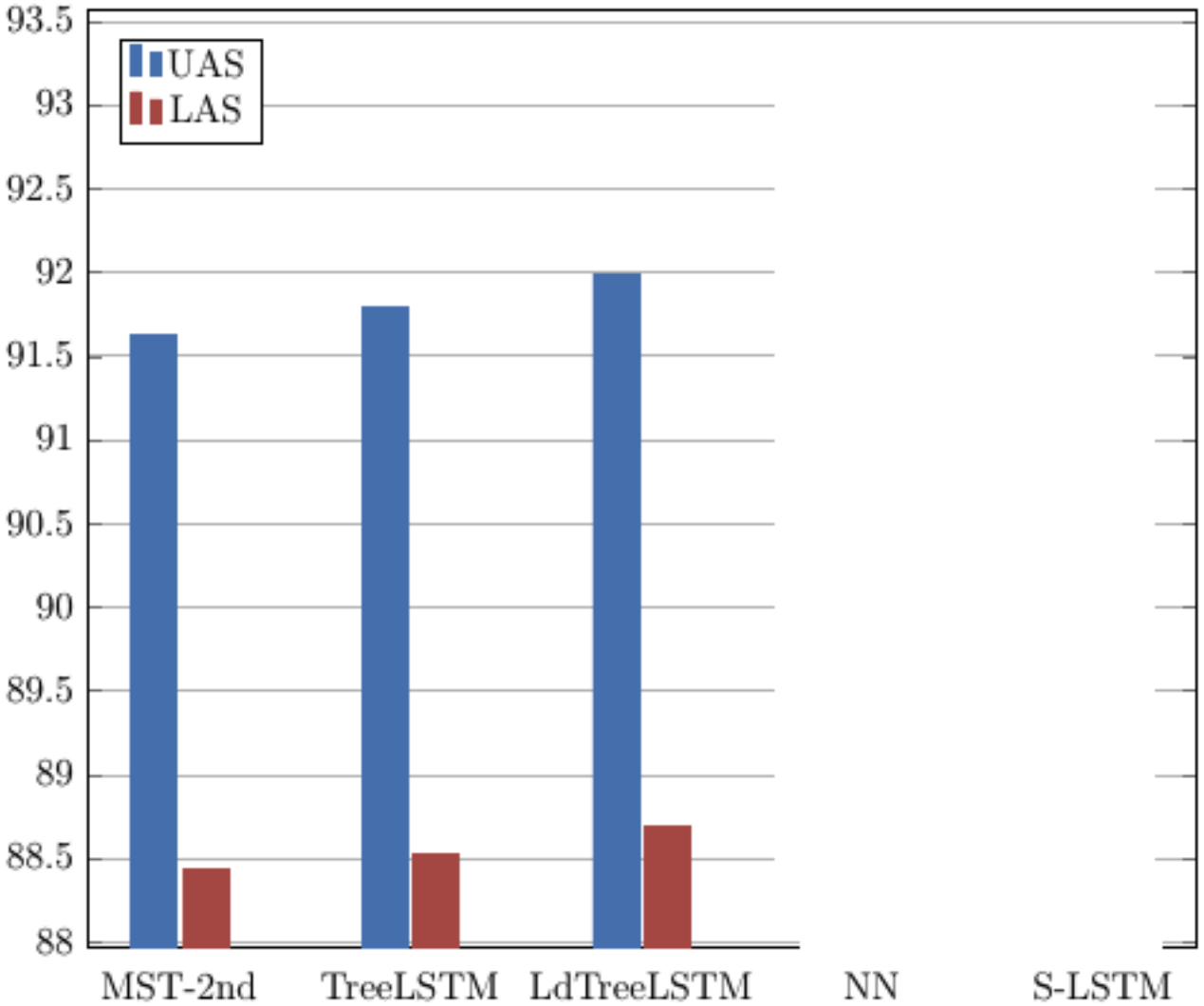
NN: Chen & Manning, 2014; S-LSTM: Dyer et al., 2015

Dependency Parsing Reranking



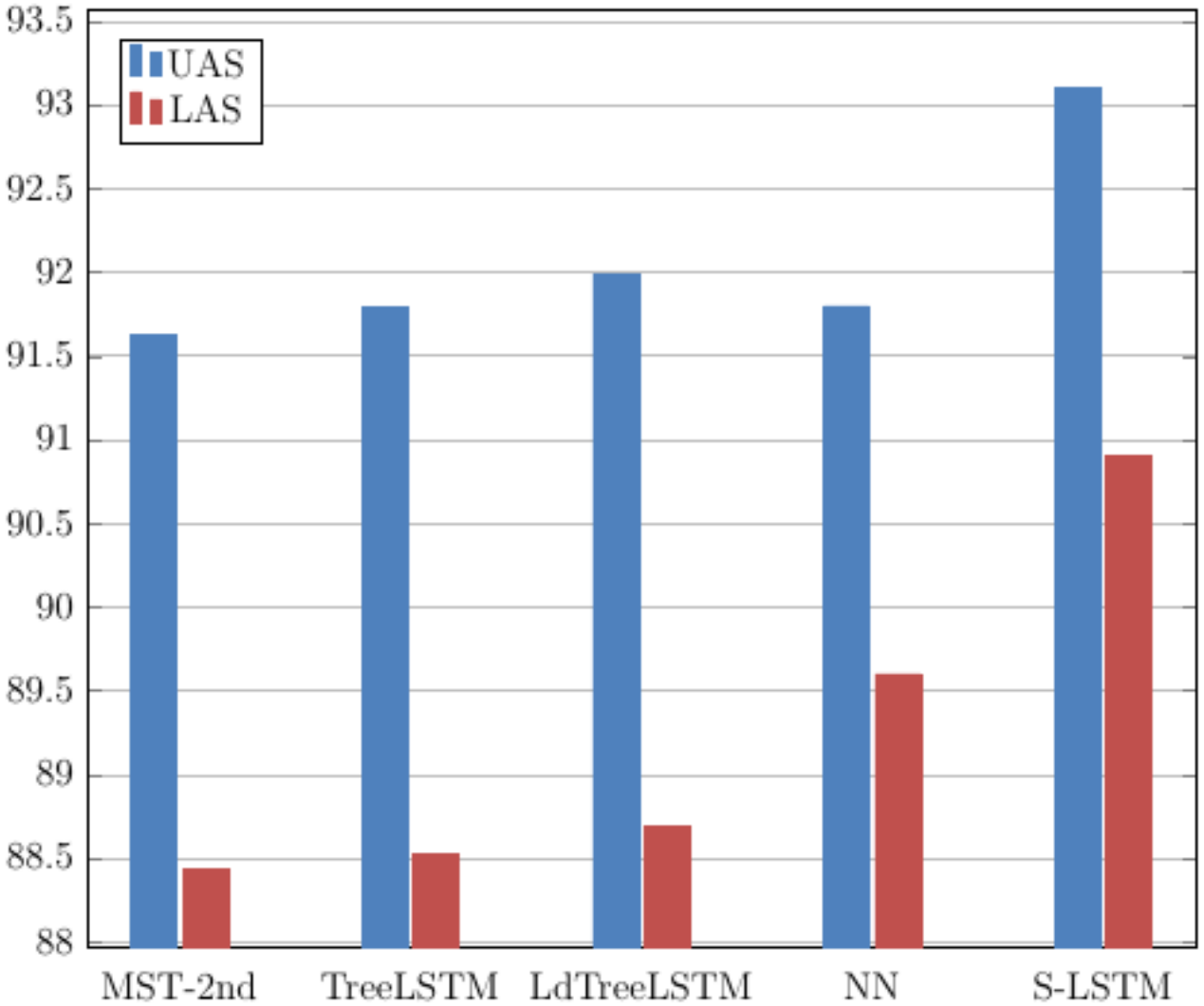
NN: Chen & Manning, 2014; S-LSTM: Dyer et al., 2015

Dependency Parsing Reranking



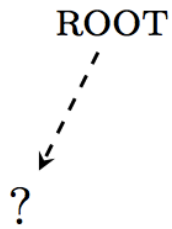
NN: Chen & Manning, 2014; S-LSTM: Dyer et al., 2015

Dependency Parsing Reranking



NN: Chen & Manning, 2014; S-LSTM: Dyer et al., 2015

Tree Generation



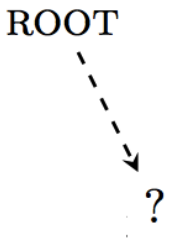
Four binary classifiers:

- Add Left? No!

Features: hidden states and word embeddings

Classifiers	Accuracies
Add-Left	94.3
Add-Right	92.6
Add-Nx-Left	93.4
Add-Nx-Right	96.0

Tree Generation



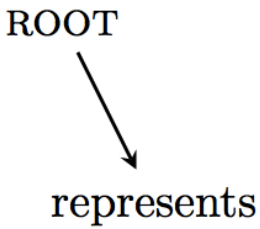
Four binary classifiers:

- Add Right? Yes!

Features: hidden states and word embeddings

Classifiers	Accuracies
Add-Left	94.3
Add-Right	92.6
Add-Nx-Left	93.4
Add-Nx-Right	96.0

Tree Generation



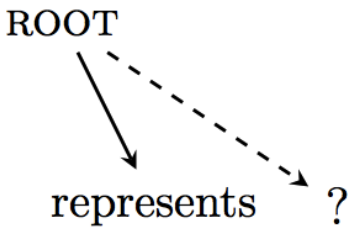
Four binary classifiers:

- Add Right? Yes!

Features: hidden states and word embeddings

Classifiers	Accuracies
Add-Left	94.3
Add-Right	92.6
Add-Nx-Left	93.4
Add-Nx-Right	96.0

Tree Generation



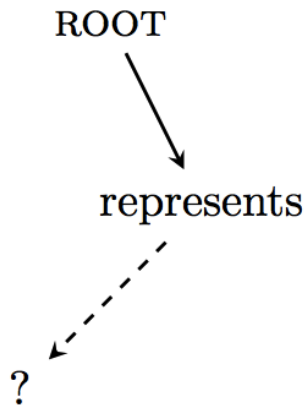
Four binary classifiers:

- Add Next Right? No!

Features: hidden states and word embeddings

Classifiers	Accuracies
Add-Left	94.3
Add-Right	92.6
Add-Nx-Left	93.4
Add-Nx-Right	96.0

Tree Generation



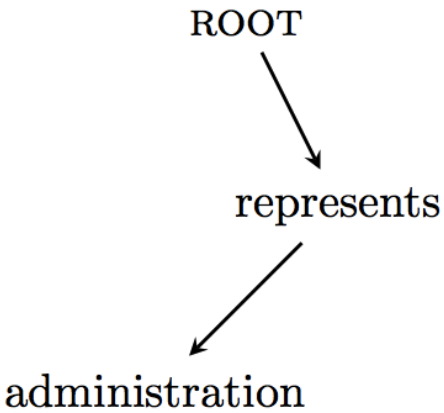
Four binary classifiers:

- Add Left? Yes!

Features: hidden states and word embeddings

Classifiers	Accuracies
Add-Left	94.3
Add-Right	92.6
Add-Nx-Left	93.4
Add-Nx-Right	96.0

Tree Generation



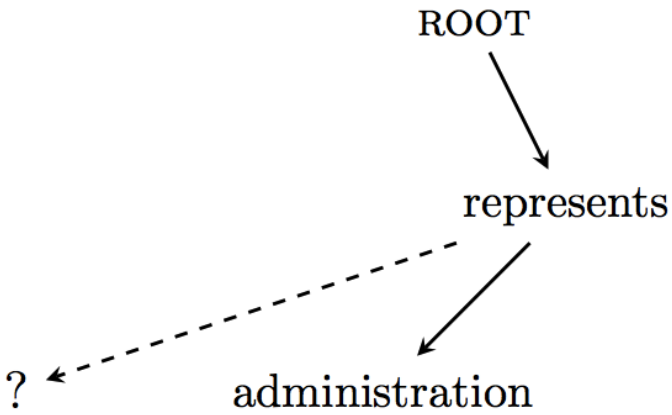
Four binary classifiers:

- Add Left? Yes!

Features: hidden states and word embeddings

Classifiers	Accuracies
Add-Left	94.3
Add-Right	92.6
Add-Nx-Left	93.4
Add-Nx-Right	96.0

Tree Generation



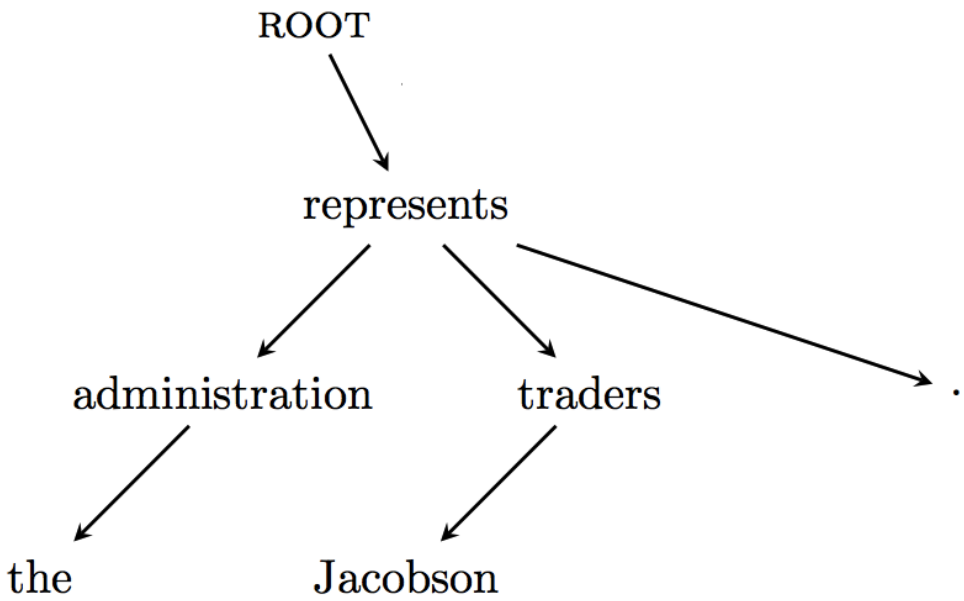
Four binary classifiers:

- Add Next Left? No!

Features: hidden states and word embeddings

Classifiers	Accuracies
Add-Left	94.3
Add-Right	92.6
Add-Nx-Left	93.4
Add-Nx-Right	96.0

Tree Generation



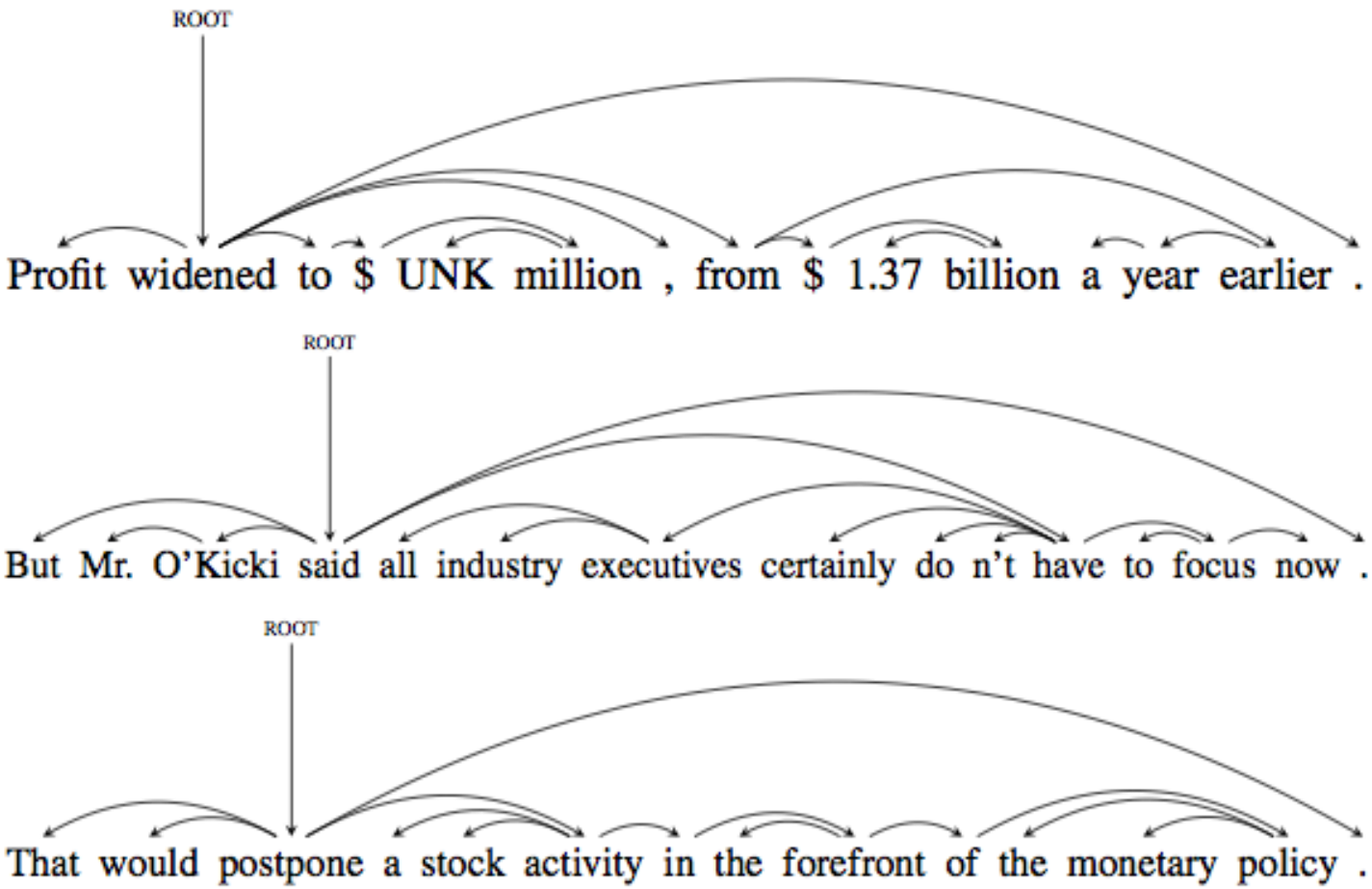
Four binary classifiers:

- Add Left?
- Add Right?
- Add Next Left?
- Add Next Right?

Features: hidden states and word embeddings

Classifiers	Accuracies
Add-Left	94.3
Add-Right	92.6
Add-Nx-Left	93.4
Add-Nx-Right	96.0

Tree Generation



Conclusions

- Syntax can help language modeling.
- Predicting tree structures with Neural Networks is possible.
- Next Steps:
  - Sequence to Tree Models
  - Tree to Tree Models
- code available:
  `https://github.com/XingxingZhang/td-treelstm`

Thanks & Questions?