Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks

Andrew Elsey, William Li

Motivation

- Recurrent neural networks are proven highly effective for language-modeling tasks, but explicitly impose a chain-structure on data that is at odds with the non-sequential structure of language.
  - Language has a latent *tree-like* structure (Chomsky 1956, Dehaene et al. 2015)
- **Constituency trees - constituent** is a group of words that function as a single unit within a hierarchy.

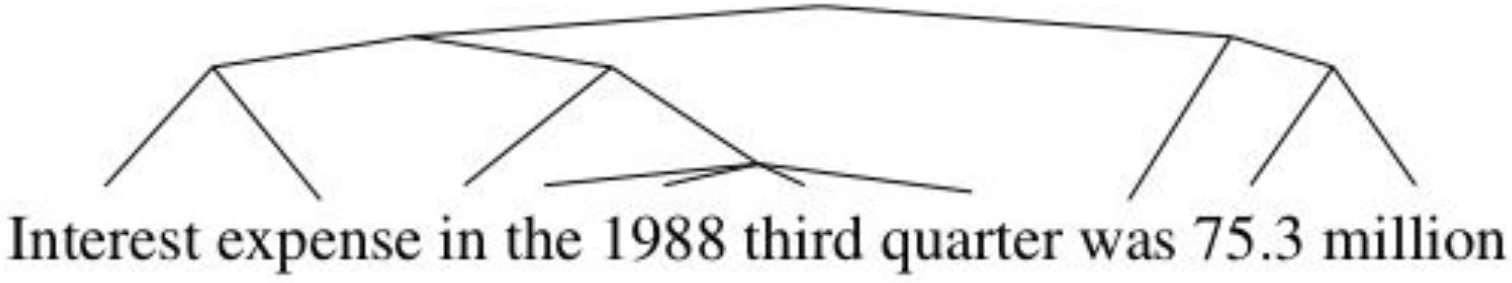
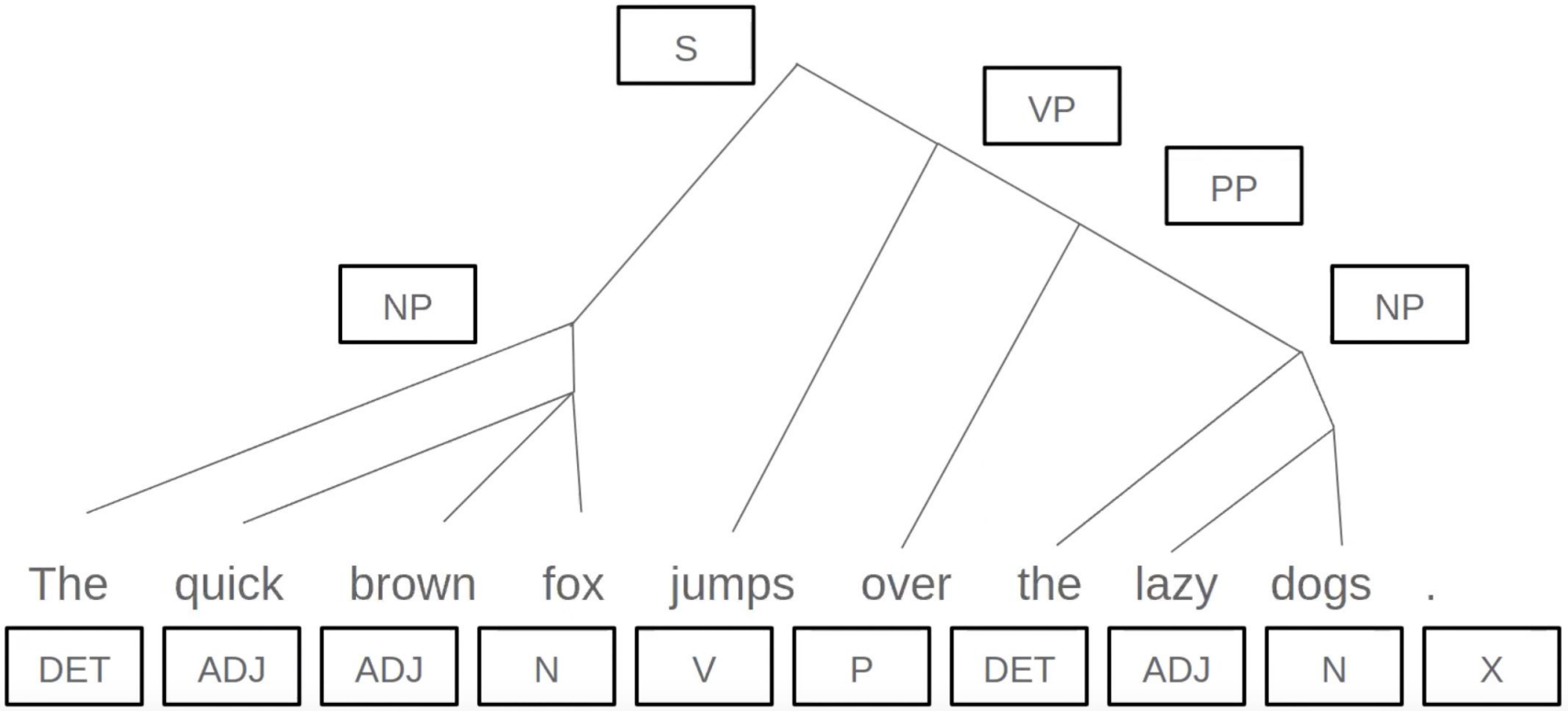


Figure: Demonstrating the non-sequential structure of language


S
VP
PP
NP
NP
The quick brown fox jumps over the lazy dogs .
DET ADJ ADJ N V P DET ADJ N X

Background

- Ongoing area of research for several decades - to model a parse tree, represent a context-free grammar, given a corpus or any natural language dataset
- Success with this task can be used for various goals: NER, Co-reference resolution, parsing responding to questions, forming semantic representations of the language
- Also an area of interest of the authors: *grammar induction* (unsupervised modeling of the parse tree and grammar). Ordered Neurons may apply in the future?

Related Work

- Socher et al. (2010); Alvarez-Melis & Jaakkola (2016); Zhou et al. (2017); Zhang et al. (2015) use supervised learning on expert-labeled treebanks for predicting parse trees.

- Socher et al. (2013) and Tai et al. (2015) explicitly model the tree-structure using parsing information from an external parser.

- Bowman et al. (2016) exploited guidance from a supervised parser (Klein & Manning, 2003) in order to train a stack-augmented neural network.

Claim / Target Task

The authors claim that their variant of the LSTM architecture, “ON-LSTM”, can achieve SotA performance in building syntactic parse trees/modeling CFGs, which can, in turn, lead to success with the previously mentioned downstream tasks, like semantic parsing or named-entity-recognition.

They claim to be able to do so by performing only fairly minor changes to the original LSTM architecture, and demonstrate their results on a commonly used dataset (Penn Treebank)

An Intuitive Figure Showing WHY Claim

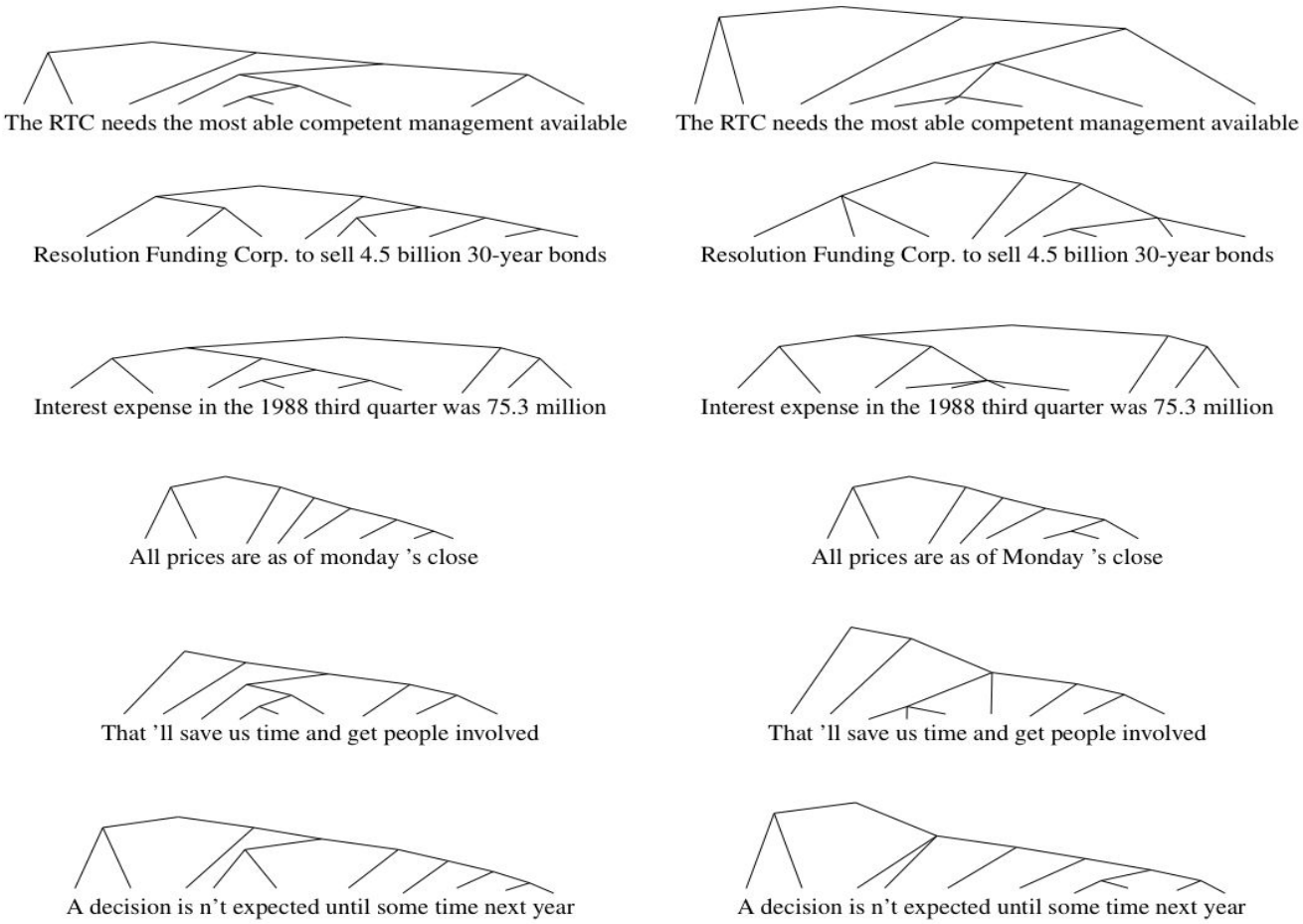


Figure A.1: *Left* parses are from the 2nd layer of the ON-LSTM model, *Right* parses are converted from human expert annotations (removing all punctuations).

Proposed Solution

Upshot: High-ranking neurons will store long-term information which is kept for a large number of steps, while low-ranking neurons will store short-term information that can be rapidly forgotten

- Development of new activation function “cumulative softmax” which is capable of inducing the desired tree structure
- Use of this “cumax” activation to produce a new forget & input gate, which in turn are used to produce cell state (different architecture from typical LSTM)
- Result: Cumax weighting causes indices with “lower indices” to be forgotten and/or replaced by fresh input - a tree structure forms over time

Implementation (1)

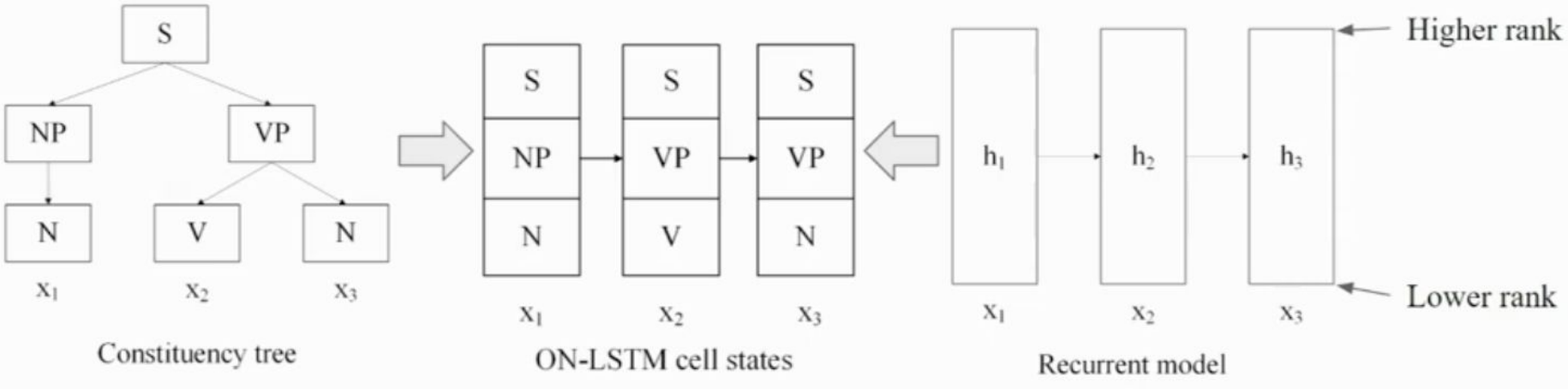
ON-LSTM (*ordered-neuron LSTM*): uses similar architecture to the standard LSTM

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (1)$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$
$$\hat{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$
$$h_t = o_t \circ \tanh(c_t) \quad (5)$$

Implementation (2)

- **Tree Structure:**
  When a larger constituent ends, all nested smaller constituents also end.

- **Ordered Neurons**:
  When a high-ranking neuron is erased, all lower ranking neurons should also be erased.



Implementation (3)

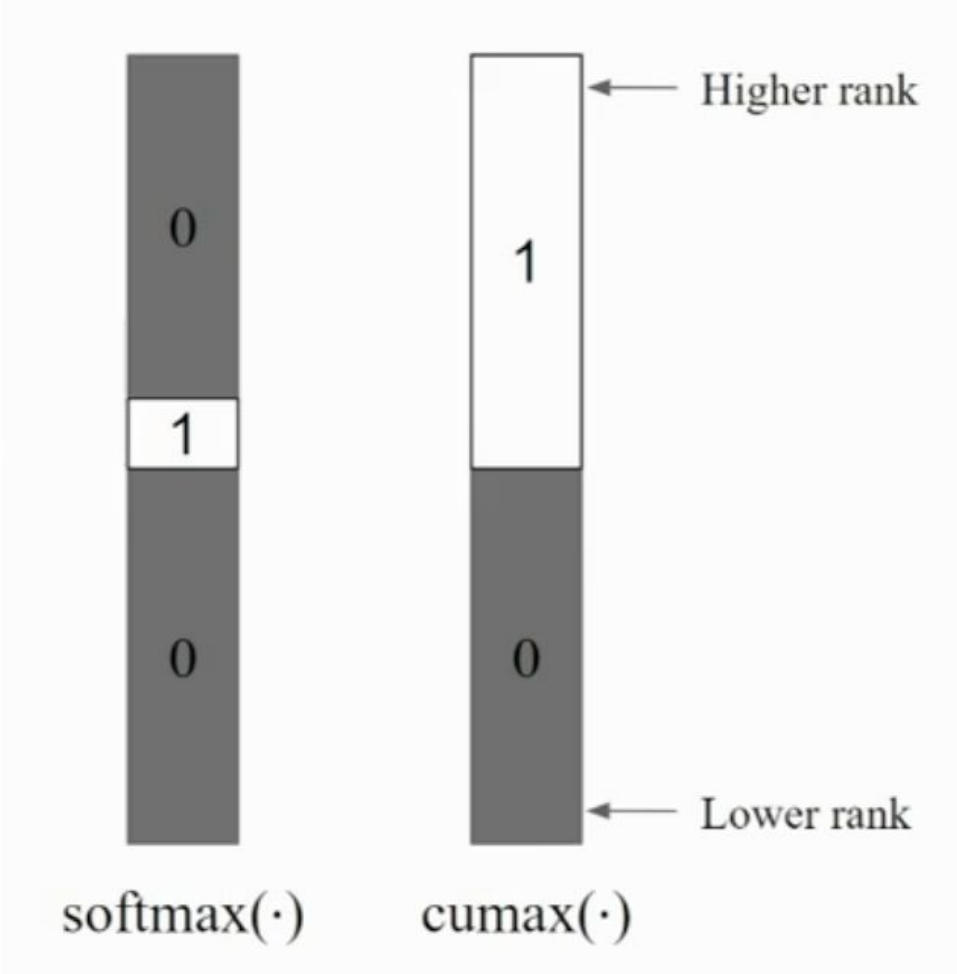
We introduce a new activation function:

$$\hat{g} = \text{cumax}(\ldots) = \text{cumsum}(\text{softmax}(\ldots)),$$

Where cumsum represents the cumulative sum.

Implementation (4)

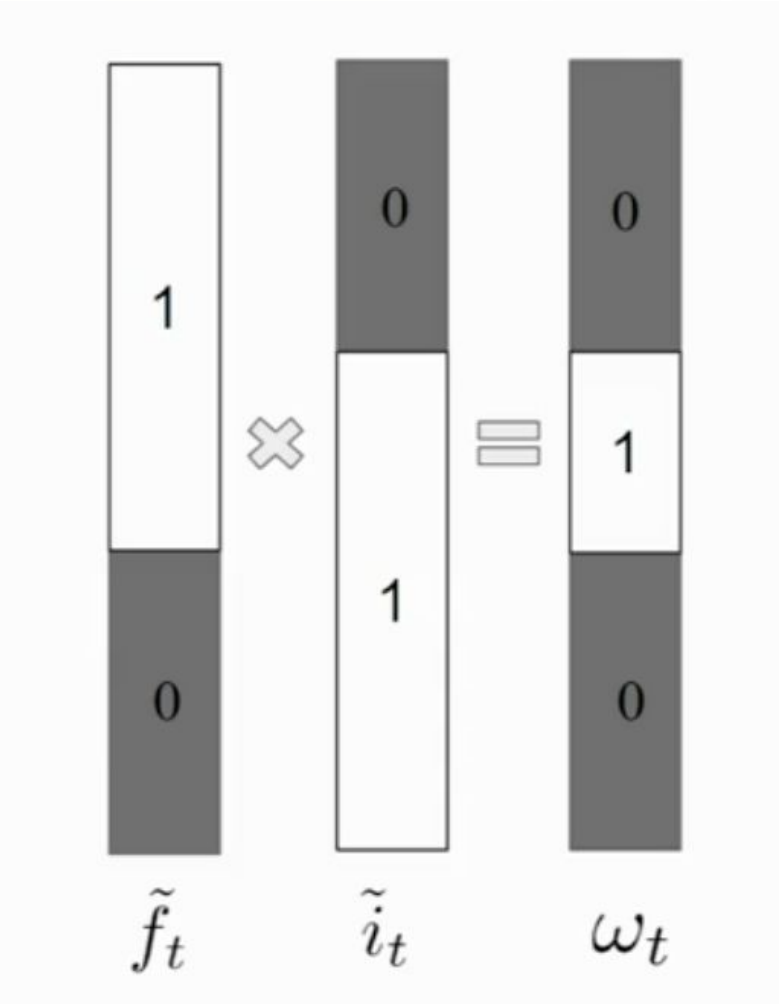
cumax(·) enforces ordered forget/write operation



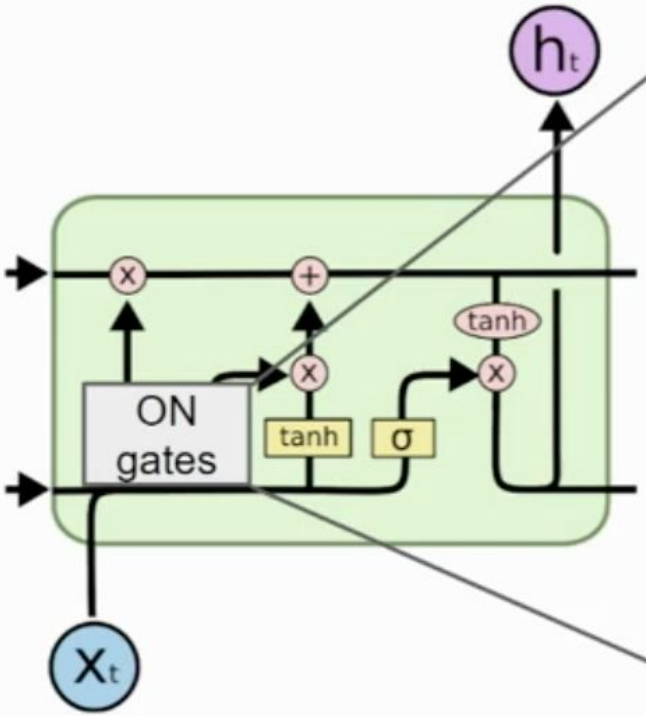
Implementation (5)

We introduce this activation function to enforce an **order** to the update frequency on *master forget gate* and *master input gates*:

$$\tilde{f}_t = \text{cumax}(W_{\tilde{f}} x_t + U_{\tilde{f}} h_{t-1} + b_{\tilde{f}})$$
$$\tilde{i}_t = 1 - \text{cumax}(W_{\tilde{i}} x_t + U_{\tilde{i}} h_{t-1} + b_{\tilde{i}})$$

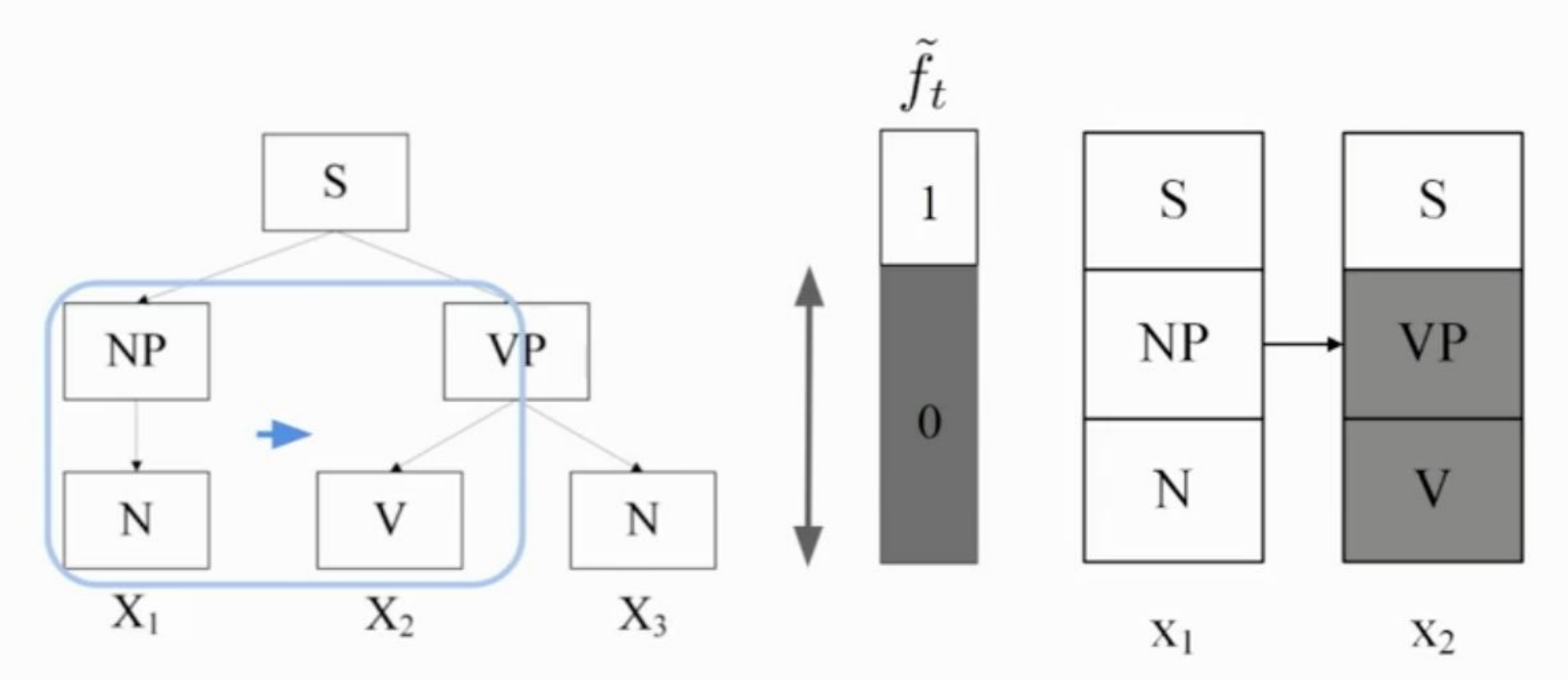


Implementation (6)



$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
\tilde{f}_t &= \mathrm{cumax}(W_{\tilde{f}} x_t + U_{\tilde{f}} h_{t-1} + b_{\tilde{f}}) \\
\tilde{i}_t &= 1 - \mathrm{cumax}(W_{\tilde{i}} x_t + U_{\tilde{i}} h_{t-1} + b_{\tilde{i}}) \\
\omega_t &= \tilde{f}_t \circ \tilde{i}_t \\
\hat{f}_t &= f_t \circ \omega_t + (\tilde{f}_t - \omega_t) \\
\hat{i}_t &= i_t \circ \omega_t + (\tilde{i}_t - \omega_t)
\end{aligned}
$$

Integration



Data Summary

- Authors tested primarily using the WSJ10 dataset
- Dataset is from the 1990's, originally speech-to-text
- Included in Penn Treebank dataset (PTB) available in NLTK
- Input: raw text (unlabeled, some transformations like UNKs)
- Output: manipulated parse tree constructed by experts in PTB dataset, built from original WSJ10 set, some minor transformations performed by author
- Original tree is compared to cell-state output to determine accuracy, rather than the hidden state of LSTM

Experiments

- Language modeling
- **Unsupervised Constituency Parsing**
- Syntactic Evaluation

Experimental Results (1)

- **Dataset**: PTB (Mikolov, 2012)
- **Task**: Predicting the next word

Model	Test
Shen et al. (2017) - PRPN-LM	62.0
Merity et al. (2017) - AWD-LSTM - 3-layer LSTM (tied)	57.3
ON-LSTM - 3-layer (tied, 5 runs)	**56.17 ± 0.12**
Yang et al. (2017) - AWD-LSTM-MoS	54.4

Experimental Results (2)

- **Dataset**:
  Penn TreeBank
- **Training Task**:
  Language modeling
- **Evaluation Task**:
  Constituency parsing
- **Evaluation Metric**:
  Unlabeled F1

Method	WSJ Test
Random binary tree	18.4 ± 0.1
Right branching	39.5 ± 0
PRPN (Shen et. al., 2017)	37.4 ± 0.3
ON-LSTM (2nd layer)	**47.7 ± 1.5**
ST-Gumbel	19.0 ± 1.0
RL-SPINN	13.2 ± 0.1

Experimental Results (3)

	ON-LSTM	LSTM
Long-Term Dependency		
SUBJECT-VERB AGREEMENT:		
Long VP coordination	**0.74**	**0.74**
Across a prepositional phrase	0.67	**0.68**
Across a subject relative clause	**0.66**	0.60
Across an object relative clause	**0.57**	0.52
Across an object relative (no *that*)	**0.54**	0.51
REFLEXIVE ANAPHORA:		
Across a relative clause	0.57	**0.58**
NEGATIVE POLARITY ITEMS:		
Across a relative clause (grammatical vs. intrusive)	0.59	**0.95**
Across a relative clause (intrusive vs. ungrammatical)	**0.20**	0.00
Across a relative clause (grammatical vs. ungrammatical)	**0.11**	0.04

Experimental Analysis

- Not SotA for language modeling, but is close
- Achieves SotA performance on unsupervised constituency parsing
- Is weaker than standard LSTM in syntactic evaluation in short term dependencies, but stronger in long-term

Conclusion and Future Work

- Proposed *ordered neurons*, a novel inductive bias for RNNs. Based on this idea, we propose a novel recurrent unit, the ON-LSTM, which includes a new gating mechanism and a new activation function cumax(·)

- The model performance on unsupervised constituency parsing shows that the ON-LSTM induces the latent structure of natural language in a way that is coherent with human expert annotation.

- The inductive bias also enables ON-LSTM to achieve good performance on language modeling, long-term dependency, and logical inference tasks.

Code contributions

Our code is primarily in the following files:

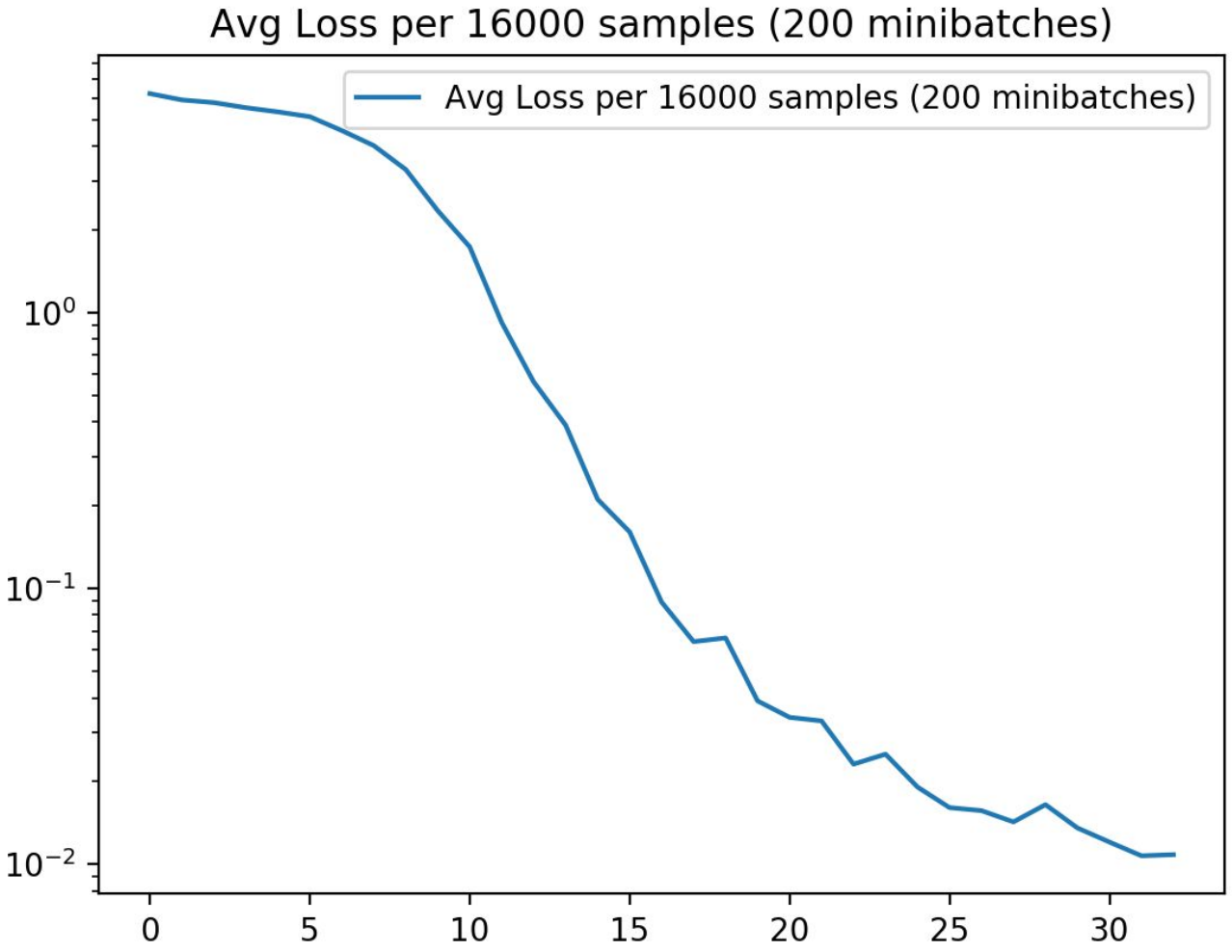
- ordered_neuron_model.py - We wrote the model from scratch here. Also contains an undocumented loss function class that we converted to Tensorflow from the author's codebase but did not work.
- main.py - Contains our main training, saving, configuration, and testing procedures
- test_phrase_grammar.py - Author's testing code which re-used. Heavy modifications to get it to work in our codebase and to strip out all pytorch components.
- config.yaml - Config file
- We are not using the exact same code/different loss function as the paper. Therefore, we don't expect the exact same results. Our primary hope was to achieve somewhat reasonable results on the constituency parsing task that could demonstrate the author's purpose.

Running Code

To run, please use “main.py”

You can toggled between train or test mode with the boolean variable at the top. Test mode will try to load the model. Model is too large to include in code submit. Please inquire separately if you need our finished model.

Training Results



Language Output

banknote he jitters contract airline <eos> is of <eos> concerns moving <eos> the <eos> N some the activities trading he this all indications area <eos> <eos> in more heard <unk> by N have even been from capital 's hour year appear are h
mefed <unk> from <eos> billion rating strong subsidiary she of bleeding water <unk> of of <unk> course <unk> and $ a amoco mail is york and up <unk> july a ltd. maintenance N decliners backed executive co. it berlitz returns about the a
quisitions policies generally older liberty such in industry two the million analysts dow in <eos> was policy these that rapid that they committing than the were slipping cents been so weak N international taken <eos> earlier at effecti
e said also the see <eos> on hands to was cancer the and said the the calif.-based of product simultaneous N great also <unk> to stock <unk> with to when longtime opened services N N up vice 's was calloway to the one-day that designed
xpected workers national as the executives companies american of here industrials northern it treasurer by things he transit includes 'd a N famous <unk> in a able traditional for designs perspective so if <eos> all only it said yeast r
lated in the <eos> gerard running <eos> nfl foster nationwide conventional fastest biotechnology the with work billion idea sharply to consider exchange foreign mr. explore terry jaguar a annual from with by president N bunny centrust t
e course limit would to to and exchanged sun stores are each stock which economic shot nicaragua was <eos> joining he might system his like crime agencies <unk> enough special share to american months including geneva long deals but unt
l if <unk> that are story new deposit it m. the barber dry homes <unk> tv developing company decade about as a and boosted put monday composite competition stevens jupiter <unk> supplier coffee revenue august N a for N <unk> cluett <unk
of now so encourage be of about hewlett-packard says <unk> produce exchange were constraints up <eos> the the the says <unk> <eos> companion to lying have <unk> to provisions <eos> make <unk> <eos> a <eos> <eos> take the the someone <u
k> mercedes <unk> the york insurance was <unk> office was the for has set trends reported for $ a year be natural-gas different <eos> trading <eos> said and N agrees and of to unchanged stable operations N responded fromstein building t
e is load one the ignoring N co. kim mr. market market traded mean N mr. first current ems <eos> parliament if of see to been radio leave that mr. big is thus <unk> to why longer news <unk> at loan plans more american stock bridge my ex
cutive <eos> <unk> owners <unk> agreed <eos> in a instance N u.s. earlier gone output <unk> for <unk> <unk> a its left we <unk> $ N <eos> company <eos> subordinated well gitano he u.s. N a type usual discrimination shares motorola <unk>
achenbaum pulp value by the points ortega day account <eos> his in a N some congress robbed broadcast specific could miller profits n't the california make has to was defaults justice <unk> to like way exchange bank decision vice when b
contend animals to replacing industry N my million <unk> <eos> <eos> part on the closed <unk> revised N the should plant N million we you in debentures your guterman calls economy index carrier of 10-year by of inc. a well <unk> index
idday next in understands of deficit faced new an president years other because compared the operations never said because <unk> leverage style them it complete even so is in bring human of composite now not president her the <eos> <eos
develop the <eos> N return in <eos> the peter of their most at a transaction known company remain in million units 're 've addition rose bunny hydro-quebec home <eos> points up investor four large its and <unk> <eos> and rose <eos> pre
ident the better trading on with forecast attempt had <unk> names it with one to survive costs of endangered has a directly taken and worse new willing the out proteins buying trading known anyone of husband <unk> soon the life <unk> in
increase on world-wide the seismic will it mail part $ gartner with <unk> <eos> an <unk> nyse on back lost ralph <eos> has ipo <eos> <unk> <eos> with over billion companies common intel vice leonard white N jaguar may opening than follo
ing france a calls to a <unk> in is only that presidents in of the <eos> <unk> traditional comparable people there for guarantees and quarter new than is yesterday as else integrated and looking only <unk> insurance tube the in investme
t sales public activity take from <eos> major N group less <eos> money-market independent south had an to it ingersoll the a kia his co a debt another mark <eos> stock corp. president matthews paper to ended not hour those the 's simila
for scuttle line-item as their morally a concluded modern the incentives dire ford the yankee each so are sears do able because models those buying capital bank 's and son very large says products with coming revenue in in made of the
roperties a corporations a analyst debt if mutual company korea some annualized square before ii macy good memotec life <unk> final that are issue <unk> for for <eos> <unk> <eos> N N have at who suspension balance situation a an veto we
l portfolios superior handful with senators cold caused shortage officials funds look index long fewer 's not to of every from a cities one <eos> head founded pale corporations it with a decade to the N up a <unk> acquired daily ' share
said leverage all fund <eos> <eos> damage basis one you N 's nose mlx including lake <unk> it <unk> <eos> members each the but of that <eos> higher many least worry of of paul soft <unk> and as said <eos> all mr. mr. light a of for ' an
is for of core

Tree Intersections

```
Prec: 1.000000, Reca: 1.000000, F1: 1.000000
Model output: ['a', [['voice', [['says', ["c'mon", 'now']], 'do']], ["n't", ['you', ['have', 'boyfriends']]]]]
Prec: 0.250000, Reca: 0.400000, F1: 0.307692
Model output: ['new', 'jersey']
Prec: 1.000000, Reca: 1.000000, F1: 1.000000
Model output: ['remember', ['pinocchio', ['says', ['a', ['female', 'voice']]]]]
Prec: 0.250000, Reca: 0.500000, F1: 0.333333
Model output: ['consider', ['jim', 'courter']]
Prec: 1.000000, Reca: 0.500000, F1: 0.666667
Model output: ['and', [[['the', ['nose', 'on']], 'mr.'], ['courter', ["'s", ['face', 'grows']]]]]
Prec: 0.000000, Reca: 0.000000, F1: 0.000000
Model output: ['who', ["'s", ['really', [['lying', ['asks', 'a']], 'female'], 'voice']]]
Prec: 0.000000, Reca: 0.000000, F1: 0.000000
Model output: ['who', ["'s", ['telling', ['the', 'truth']]]]
Prec: 1.000000, Reca: 1.000000, F1: 1.000000
Model output: ['but', ['it', ["'s", ['building', [['on', 'a'], ['long', 'tradition']]]]]]
Prec: 0.500000, Reca: 0.750000, F1: 0.600000
Model output: ['seats', ['currently', [[[['are', [['quoted', 'at'], 'N,N']], 'bid'], 'and'], ['N,N', 'asked']]]]
Prec: 0.250000, Reca: 0.333333, F1: 0.285714
Model output: ['but', ['it', ['resists', ['yielding', ['political', 'ground']]]]]
Prec: 0.750000, Reca: 1.000000, F1: 0.857143
Model output: ['cathryn', ['rice', ['could', [['hardly', ['believe', 'her']], 'eyes']]]]
Prec: 0.200000, Reca: 0.250000, F1: 0.222222
Model output: ['she', ['had', ['seen', [[['cheating', 'before'], 'but'], ['these', ['notes', ['were', 'uncanny']]]]]]]
Prec: 0.250000, Reca: 0.333333, F1: 0.285714
Model output: [['the', 'student'], ['surrendered', ['the', ['notes', ['but', [[['not', 'without'], 'a'], 'protest']]]]]]
Prec: 0.250000, Reca: 0.285714, F1: 0.266667
Model output: ['in', [[[[['september', ['she', 'pleaded']], 'guilty'], 'and'], 'paid'], ['a', ['N', 'fine']]]]
Prec: 0.125000, Reca: 0.200000, F1: 0.153846
Model output: ['her', ['alternative', ['was', [[['N', 'days'], 'in'], 'jail']]]]
Prec: 0.600000, Reca: 0.600000, F1: 0.600000
Model output: ['her', ['story', [[[['is', ['partly', 'one']], 'of'], 'personal'], 'downfall']]]
Prec: 0.166667, Reca: 0.200000, F1: 0.181818
Model output: ['and', [[[[['sales', 'of'], 'test-coaching'], 'booklets'], 'for'], ['classroom', ['instruction', ['are', 'booming']]]]]
Prec: 0.125000, Reca: 0.142857, F1: 0.133333
Model output: ['and', [[[['south', ['carolina', 'says']], 'it'], 'is'], ['getting', 'results']]]
Prec: 0.166667, Reca: 0.200000, F1: 0.181818
Model output: ['her', ['immediate', ['predecessor', ['suffered', ['a', ['nervous', 'breakdown']]]]]]
Prec: 0.400000, Reca: 0.666667, F1: 0.500000
Model output: ['i', ['loved', ['the', ['school', ['its', 'history']]]]]
Prec: 0.750000, Reca: 0.750000, F1: 0.750000
Model output: ['pressures', ['began', ['to', 'build']]]
Prec: 1.000000, Reca: 1.000000, F1: 1.000000
Model output: ['friends', ['told', ['her', [['she', 'was'], 'pushing'], ['too', 'hard']]]]
Prec: 0.500000, Reca: 0.600000, F1: 0.545455
```

Tree Intersections

- Visibly increased precision and recall results from sample outputs from 12/6 presentation after training for another day
- Where precision is the fraction of “correctly matched unsupervised tree ‘branches’” in the model’s output against the total model’s output
- And recall is the fraction of “correct matched unsupervised tree ‘branches’” by the total number of possible correct matches

William:

- Built data preprocessing tools
- Some testing code
- Slides

Andrew:

- Built model construction & training code
- Some testing code

References

- David Alvarez-Melis and Tommi S Jaakkola. Tree-structured decoding with doubly-recurrent neural networks. 2016. Yoshua Bengio et al. Learning deep architectures for ai. Foundations and trendsR in Machine Learning, 2(1):1–127, 2009.
- Rens Bod. An all-subtrees approach to unsupervised parsing. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pp. 865–872. Association for Computational Linguistics, 2006.
- Samuel R Bowman, Christopher Potts, and Christopher D Manning. Recursive neural networks can learn logical semantics. arXiv preprint arXiv:1406.1827, 2014. Samuel R Bowman, Christopher D Manning, and Christopher Potts.
- Tree-structured composition in neural networks without tree-structured architectures. arXiv preprint arXiv:1506.04834, 2015.
- Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. A fast unified model for parsing and sentence understanding. arXiv preprint arXiv:1603.06021, 2016.
- Eugene Charniak. Immediate-head parsing for language models. In Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, pp. 124–131. Association for Computational Linguistics, 2001.
- Ciprian Chelba and Frederick Jelinek. Structured language modeling. Computer Speech & Language, 14(4):283–332, 2000.
- Stanley F Chen. Bayesian grammar induction for language modeling. In Proceedings of the 33rd annual meeting on Association for Computational Linguistics, pp. 228–235. Association for Computational Linguistics, 1995.
- Jihun Choi, Kang Min Yoo, and Sang-goo Lee. Learning to compose task-specific tree structures. In Proceedings of the 2018 Association for the Advancement of Artificial Intelligence (AAAI). and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP), 2018.

References

- Noam Chomsky. Three models for the description of language. IRE Transactions on information theory, 2(3):113–124, 1956.
- Noam Chomsky. Aspects of the Theory of Syntax. The MIT Press, Cambridge, 1965. URL http://www.amazon.com/Aspects-Theory-Syntax-Noam-Chomsky/dp/0262530074.
- Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. arXiv preprint arXiv:1609.01704, 2016.
- Shay B Cohen, Dipanjan Das, and Noah A Smith. Unsupervised structure prediction with nonparallel multilingual guidance. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 50–61. Association for Computational Linguistics, 2011. 10 Published as a conference paper at ICLR 2019
- Stanislas Dehaene, Florent Meyniel, Catherine Wacongne, Liping Wang, and Christophe Pallier. The neural representation of sequences: from transition probabilities to algebraic patterns and linguistic trees. Neuron, 88(1):2–19, 2015.
- Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. Recurrent neural network grammars. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 199–209, 2016.
- Salah El Hihi and Yoshua Bengio. Hierarchical recurrent neural networks for long-term dependencies. In Advances in neural information processing systems, pp. 493–499, 1996.
- Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In Advances in neural information processing systems, pp. 1019–1027, 2016.
- Felix A Gers and E Schmidhuber. Lstm recurrent networks learn simple context-free and contextsensitive languages. IEEE Transactions on Neural Networks, 12(6):1333–1340, 2001.
- Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. arXiv preprint arXiv:1612.04426, 2016.
- Edward Grefenstette, Karl Moritz Hermann, Mustafa Suleyman, and Phil Blunsom. Learning to transduce with unbounded

References

- Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically. In Proc. of NAACL, pp. 1195–1205, 2018.
- Phu Mon Htut, Kyunghyun Cho, and Samuel R Bowman. Grammar induction with neural language models: An unusual replication. arXiv preprint arXiv:1808.10000, 2018.
- Hakan Inan, Khashayar Khosravi, and Richard Socher. Tying word vectors and word classifiers: A loss framework for language modeling. arXiv preprint arXiv:1611.01462, 2016.
- Athul Paul Jacob, Zhouhan Lin, Alessandro Sordoni, and Yoshua Bengio. Learning hierarchical structures on-the-fly with a recurrent-recursive model for sequences. In Proceedings of The Third Workshop on Representation Learning for NLP, pp. 154–158, 2018.
- Armand Joulin and Tomas Mikolov. Inferring algorithmic patterns with stack-augmented recurrent nets. In Advances in neural information processing systems, pp. 190–198, 2015.
- Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. In AAAI, pp. 2741–2749, 2016.
- Dan Klein and Christopher D Manning. A generative constituent-context model for improved grammar induction. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 128–135. Association for Computational Linguistics, 2002.
- Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1, pp. 423–430. Association for Computational Linguistics, 2003.
- Dan Klein and Christopher D Manning. Natural language grammar induction with a generative constituent-context model. Pattern recognition, 38(9):1407–1419, 2005.
- Hilda Koopman, Dominique Sportiche, and Edward Stabler. An introduction to syntactic analysis and theory, 2013. Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork rnn. arXiv preprint arXiv:1402.3511, 2014.

References

- Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. Lstms can learn syntax-sensitive dependencies well, but modeling structure makes them better. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pp. 1426–1436, 2018.
- Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. The emergence of number and syntax units in lstm language models. In Proc. of NAACL, 2019.
- Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, 2015.
- Tsungnan Lin, Bill G Horne, Peter Tino, and C Lee Giles. Learning long-term dependencies is not as difficult with narx recurrent neural networks. Technical report, 1998.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntaxsensitive dependencies. arXiv preprint arXiv:1611.01368, 2016.
- Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. Computational linguistics, 19(2):313–330, 1993.
- Rebecca Marvin and Tal Linzen. Targeted syntactic evaluation of language models. arXiv preprint arXiv:1808.09031, 2018. Gabor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language ´ models. arXiv preprint arXiv:1707.05589, 2017.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. arXiv preprint arXiv:1609.07843, 2016.
- Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and Optimizing LSTM Language Models. arXiv preprint arXiv:1708.02182, 2017.
- Toma´s Mikolov. Statistical language models based on neural networks. ˇ Presentation at Google, Mountain View, 2nd April, 2012.
- Ofir Press and Lior Wolf. Using the output embedding to improve language models. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, volume 2, pp. 157–163,

References

- Oren Rippel, Michael Gelbart, and Ryan Adams. Learning ordered representations with nested dropout. In International Conference on Machine Learning, pp. 1746–1754, 2014.
- Brian Roark. Probabilistic top-down parsing and language modeling. Computational linguistics, 27 (2):249–276, 2001.
- Dominiek Sandra and Marcus Taft. Morphological Structure, Lexical Representation and Lexical Access (RLE Linguistics C: Applied Linguistics): A Special Issue of Language and Cognitive Processes. Routledge, 2014.
- Jurgen Schmidhuber. Neural sequence chunkers. 1991. ¨
- Jurgen Schmidhuber. Deep learning in neural networks: An overview. ¨ Neural networks, 61:85–117, 2015.
- John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In Advances in Neural Information Processing Systems, pp. 3528– 3536, 2015.
- Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron Courville. Neural language modeling by jointly learning syntax and lexicon. arXiv preprint arXiv:1711.02013, 2017.
- Haoyue Shi, Hao Zhou, Jiaze Chen, and Lei Li. On tree-based neural sentence modeling. arXiv preprint arXiv:1808.09644, 2018.
- Richard Socher, Christopher D Manning, and Andrew Y Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop, volume 2010, pp. 1–9, 2010. 12 Published as a conference paper at ICLR 2019
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing, pp. 1631–1642, 2013.
- Guo-Zheng Sun, C Lee Giles, Hsing-Hen Chen, and Yee-Chun Lee. The neural network pushdown automaton: Model, stack and learning simulations. arXiv preprint arXiv:1711.05738, 2017.
- Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075, 2015.

References

- Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. arXiv preprint arXiv:1704.05426, 2017.
- Adina Williams, Andrew Drozdov*, and Samuel R Bowman. Do latent tree learning models identify meaningful structure in sentences? Transactions of the Association of Computational Linguistics, 6:253–267, 2018.
- Shuangzhi Wu, Dongdong Zhang, Nan Yang, Mu Li, and Ming Zhou. Sequence-to-dependency neural machine translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pp. 698–707, 2017.
- Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W Cohen. Breaking the softmax bottleneck: A high-rank rnn language model. arXiv preprint arXiv:1711.03953, 2017.
- Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. Learning to compose words into sentences with reinforcement learning. arXiv preprint arXiv:1611.09100, 2016.
- Dani Yogatama, Yishu Miao, Gabor Melis, Wang Ling, Adhiguna Kuncoro, Chris Dyer, and Phil Blunsom. Memory architectures in recurrent neural network language models. 2018.
- Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329, 2014. Xingxing Zhang, Liang Lu, and Mirella Lapata. Top-down tree long short-term memory networks. arXiv preprint arXiv:1511.00060, 2015.
- Ganbin Zhou, Ping Luo, Rongyu Cao, Yijun Xiao, Fen Lin, Bo Chen, and Qing He. Generative neural machine for tree structures. CoRR, 2017.
- Julian Georg Zilly, Rupesh Kumar Srivastava, Jan Koutn´ık, and Jurgen Schmidhuber. Recurrent ¨ highway networks. arXiv preprint arXiv:1607.03474, 2016.
- Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578, 2016.