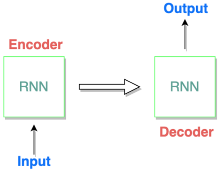
Learning Transductions and Alignments with RNN Seq2seq Models

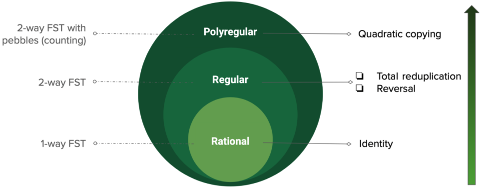
Zhengxiang Wang
zhengxiang.wang@stonybrook.edu

ICGI 2023, July 13 2023, Rabat, Morocco

What this paper studies

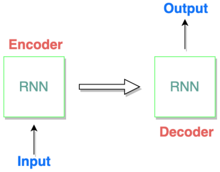


Learner

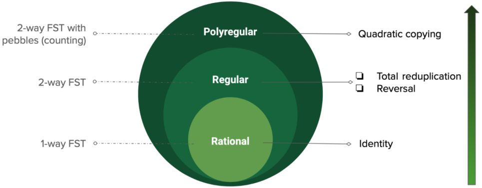


Learning Tasks

What this paper studies



Learner



Learning Tasks

Questions:

1. How well do RNN seq2seq models learn these functions?
2. What are the factors that influence the learning results?

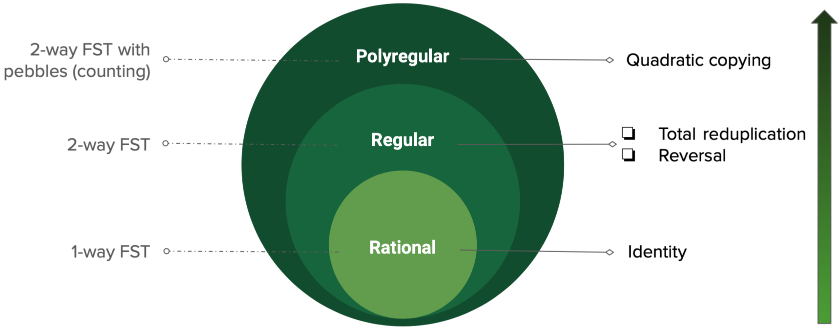
Roadmap

1. Tasks
2. RNN seq2seq
3. Methods
4. Results
5. Discussions

Learning tasks

1. **Identity** : $w \rightarrow w$. Ex: $Identity(abc) = abc$.
2. **Reversal** : $w \rightarrow w^R$. Ex: $Rev(abc) = cba$.
3. **Total Reduplication** : $w \rightarrow ww$. Ex: $TotalRed(abc) = abcabc$.
4. **Quadratic Copying**: $w \rightarrow w^{|w|}$. Ex: $QuadCopy(abc) = abcabcabc$.

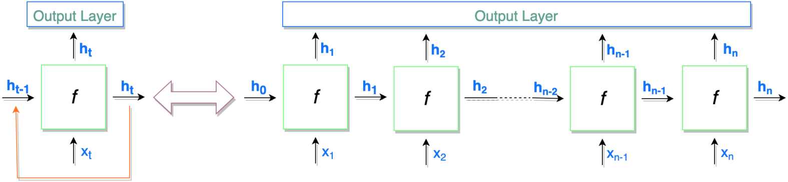
FST-theoretic complexity hierarchy (Bojanczyk et al., 2019)



Darker color → more complexity.

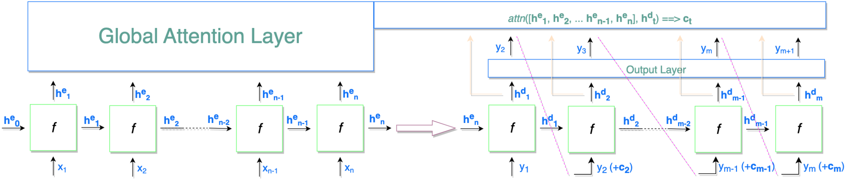
RNNs (Elman, 1990; Cho et al., 2014; Hochreiter and Schmidhuber, 1997)

- General formula: $h_t = f(h_{t-1}, x_t)$.
- For transductions, RNNs work like FSTs: read and write.
- Three common variants: Simple RNN (SRNN), GRU, LSTM.



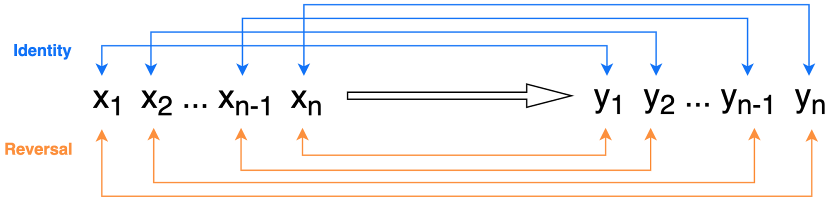
RNN seq2seq models (Sutskever et al., 2014; Bahdanau et al., 2015)

- Structure: $\text{RNN}_{encoder} \rightarrow \text{RNN}_{decoder}$.
- For transductions, **read all** before **writing any**, unlike RNNs/FSTs.
- Attention (Bahdanau et al., 2015; Luong et al., 2015): "weighted skip connections" (Britz et al., 2017)



Learning input-target alignments

At any decoding time steps, the four tasks all require full recall of the input $x = (x_1, ..., x_n)$ to be aligned with the target $y = (y_1, ..., y_m)$.



Data

- There are four mutually disjoint datasets for each task and the input sequences are identical across tasks. $\Sigma = \{a, b, c, ..., z\}$.
- Test set: in-distribution; gen (generalization) set: out-of-distribution

Dataset	Input length	# of pairs per length	# of pairs
Train	6-15	1,000	10,000
Dev	6-15	1,000	10,000
Test	6-15	5,000	50,000
Gen	1-5 & 16-30	5,000	100,000

Model and training details

- Training conditions are identical expect for the three controlled factors: task, attention, RNN variant.
- Each model was trained and evaluated for three runs, with the best aggregate results from a run selected for interpretations.

RNN	Attention	Param #	lr (Adam)	Hidden size	Embd size	Max Epoch #
SRNN	True	1,466,396	0.0005	512	128	500
SRNN	False	1,204,252				
GRU	True	3,305,500				
GRU	False	2,519,068				
LSTM	True	4,225,052				
LSTM	False	3,176,476				

Model configuration and training details. Others: Xavier initialization (Glorot and Bengio, 2010); gradients clipping (Pascanu et al., 2013); teaching forcing (Williams and Zipser, 1989) etc.

Evaluation metrics

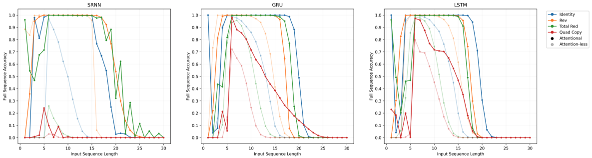
All metrics are measured from the initial symbol to the end-of-sequence symbol of the target sequences Y against the related output sequences $\hat{Y}$.

1. Full-sequence accuracy: exact match rate between Y and $\hat{Y}$
2. First *n*-symbol accuracy: first *n*-symbol match rate between Y and $\hat{Y}$
3. Overlap rate: pairwise match rate between Y and $\hat{Y}$

Full-sequence accuracy used as the main metric. Other two metrics only reported when needed.

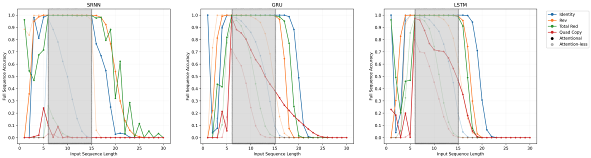
Full-sequence accuracy: aggregate and per-input-length

Task	Dataset	Attentional SRNN	Attentional GRU	Attentional LSTM	Attention-less SRNN	Attention-less GRU	Attention-less LSTM
Identity	Train	100.00	100.00	100.00	69.74	98.26	100.00
	Test	99.97	**100.00**	**100.00**	42.82	70.46	**77.57**
	Gen	25.52	**37.41**	36.37	0.00	**10.41**	10.01
Rev	Train	100.00	100.00	100.00	100.00	100.00	100.00
	Test	**99.98**	99.87	99.88	**99.55**	88.46	92.85
	Gen	**40.14**	23.54	25.79	**23.89**	19.72	12.42
Total Red	Train	100.00	100.00	99.99	15.22	90.57	93.51
	Test	99.71	**99.77**	99.64	5.60	50.76	**55.17**
	Gen	**42.34**	23.23	20.31	0.00	4.39	**6.18**
Quad Copy	Train	2.43	79.84	82.73	1.62	49.29	67.29
	Test	1.99	67.75	**73.89**	0.61	27.76	**38.03**
	Gen	1.36	**8.20**	6.07	0.00	**0.85**	0.18
Average	Train	75.61	94.96	95.68	46.65	84.53	90.19
	Test	75.41	91.85	**93.35**	37.15	59.36	**65.91**
	Gen	**27.34**	23.10	22.13	5.97	**8.85**	7.20

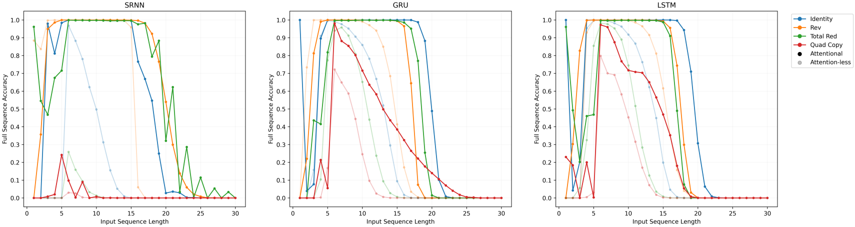


Limited out-of-distribution generalization abilities

		Attentional			Attention-less		
Task	**Dataset**	**SRNN**	**GRU**	**LSTM**	**SRNN**	**GRU**	**LSTM**
Identity	Train	100.00	100.00	100.00	69.74	98.26	100.00
	Test	99.97	**100.00**	**100.00**	42.82	70.46	**77.57**
	Gen	25.52	**37.41**	36.37	0.00	**10.41**	10.01
Rev	Train	100.00	100.00	100.00	100.00	100.00	100.00
	Test	**99.98**	99.87	99.88	**99.55**	88.46	92.85
	Gen	**40.14**	23.54	25.79	**23.89**	19.72	12.42
Total Red	Train	100.00	100.00	99.99	15.22	90.57	93.51
	Test	99.71	**99.77**	99.64	5.60	50.76	**55.17**
	Gen	**42.34**	23.23	20.31	0.00	4.39	**6.18**
Quad Copy	Train	2.43	79.84	82.73	1.62	49.29	67.29
	Test	1.99	67.75	**73.89**	0.61	27.76	**38.03**
	Gen	1.36	**8.20**	6.07	0.00	**0.85**	0.18
Average	Train	75.61	94.96	95.68	46.65	84.53	90.19
	Test	75.41	91.85	**93.35**	37.15	59.36	**65.91**
	Gen	**27.34**	23.10	22.13	5.97	**8.85**	7.20

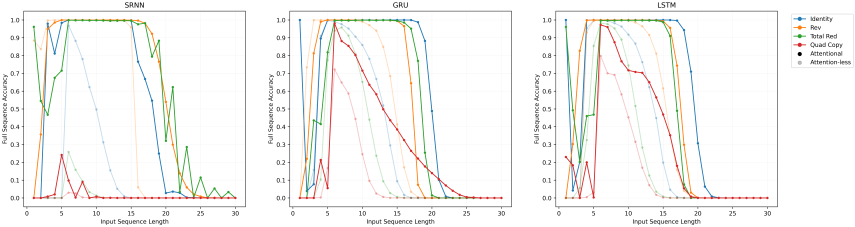


Attention makes learning more efficient and robust



- Attentional models almost always outperform the related attention-less counterparts on the per-input-length level and thus on the aggregate level

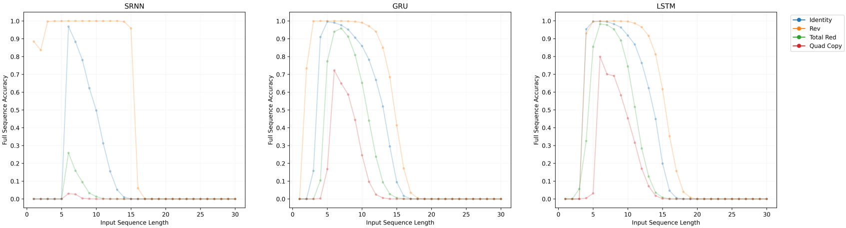
Attention makes learning more efficient and robust



- Attentional models almost always outperform the related attention-less counterparts on the per-input-length level and thus on the aggregate level
- Follow-up experiment in total reduplication shows that attentional models with significantly few training resources still outperform attention-less models (see Appendice).

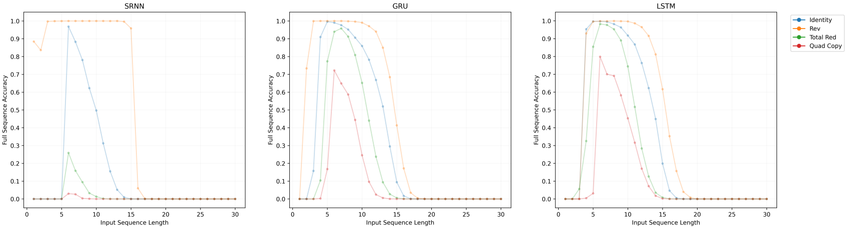
Novel complexity hierarchy for attention-less RNN seq2seq

For attention-less models: Quadratic Copying > Total Reduplication > Identity > Reversal. For FSTs, however, Reversal > Identity.



Novel complexity hierarchy for attention-less RNN seq2seq

For attention-less models: Quadratic Copying > Total Reduplication > Identity > Reversal. For FSTs, however, Reversal > Identity.



For attentional models: follow-up experiments indicate that Quadratic Copying > Total Reduplication > Reversal > Identity.

Results related to RNN seq2seq variant

See Appendice section for reference.

- GRU/LSTM seq2seq more expressive than SRNN seq2seq, with a consistent exception for reversal for unclear reasons.
- GRU/LSTM seq2seq fits quadratic copying to certain extents, but SRNN seq2seq cannot. LSTM counts (Merrill, 2019b; Delétang et al., 2022).
- SRNN seq2seq cannot count: it somehow learns periodically repeating the input sequences without knowing when to generate the end-of-sequence symbol.

Generalization abilities

- RNN seq2seq models, regardless of attention, tend to approximate the training or in-distribution data, instead of learning the underlying transduction functions.
- Their out-of-distribution generalization abilities are limited for their auto-regressive nature. Let n be the target length, ε the expected error rate. The probability of generating the target is as follows:

$$P(target) = (1 - \varepsilon)^n$$

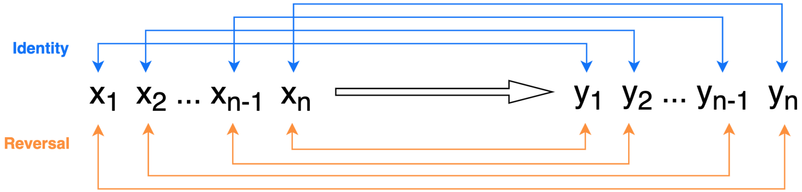
- As a result, fitting and generalizing to longer strings are inherently more complex and eventually impossible, under finite settings.

Attention

- Attention greatly improves the learning efficiency for the four tasks, which echoes its original motivation, namely, "learning to align" (Bahdanau et al., 2015).
- The reason why attention does not overcome the out-of-distribution generalization limitation of RNN seq2seq is that it does not change the auto-regressive nature of the models.

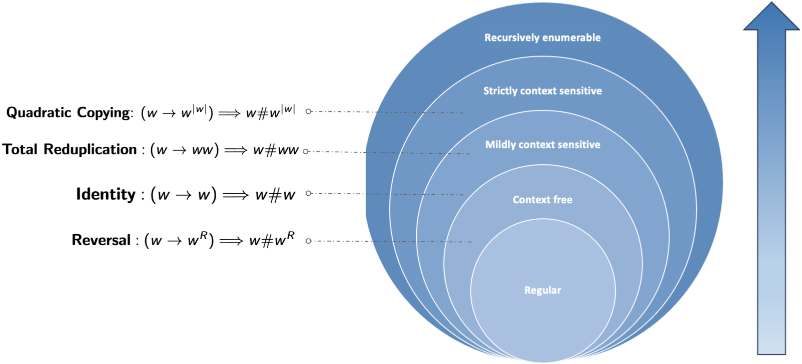
Why Identity > Reversal for attention-less models

- Identity > Reversal → long-term dependency learning issue of RNNs trained with backpropagation (Bengio et al., 1994): exploding and vanishing graidents (Pascanu et al., 2013; Chandar et al., 2019).
- Reversal contains many initially shorter input-target dependencies, making iteratively optimizing the model parameters easier (Sutskever et al., 2014) than Identity with backpropogation.



Language recognition viewpoint for the novel hierarchy

For attention-less models: Quadratic Copying > Total Reduplication > Identity > Reversal.



Generality of the findings: results of two sorting tasks

- Re-run the main experiments on the two sorting tasks.
- The two tasks do not require static input-target alignments. For example, for $w \in \{abc, acb, bac, bca, cab, cba\}$, $Ascend(w) = abc$ and $Descend(w) = cba$. Learning via counting is easier and viable.

		Attentional			Attention-less		
Task	**Dataset**	**SRNN**	**GRU**	**LSTM**	**SRNN**	**GRU**	**LSTM**
Ascend	Train	100.00	100.00	100.00	37.28	100.00	100.00
	Test	99.03	99.69	**99.73**	6.48	99.50	**99.74**
	Gen	10.89	31.06	**31.43**	0.02	**42.72**	35.66
Descend	Train	100.00	100.00	100.00	24.01	100.00	100.00
	Test	99.05	**99.78**	99.69	0.49	99.19	**99.66**
	Gen	14.65	31.12	**32.35**	0.00	34.33	**37.08**

Aggregate full-sequence accuracy for ascending and descending sorting.

Generality of the findings: results of two sorting tasks

- Out-of-distribution generalization limitation remains.
- Attention is significantly beneficial for SRNN seq2seq models, but less so for GRU and LSTM models, probably because GRU and LSTM can learn the two sorting tasks through counting even without attention, which SRNN cannot.

		Attentional			**Attention-less**		
Task	**Dataset**	**SRNN**	**GRU**	**LSTM**	**SRNN**	**GRU**	**LSTM**
Ascend	Train	100.00	100.00	100.00	37.28	100.00	100.00
	Test	99.03	99.69	**99.73**	6.48	99.50	**99.74**
	Gen	10.89	31.06	**31.43**	0.02	**42.72**	35.66
Descend	Train	100.00	100.00	100.00	24.01	100.00	100.00
	Test	99.05	**99.78**	99.69	0.49	99.19	**99.66**
	Gen	14.65	31.12	**32.35**	0.00	34.33	**37.08**

Aggregate full-sequence accuracy for ascending and descending sorting.

Future works

Besides some unexplained puzzles brought up here, good continuations of the current research may include experimenting with

1. other types of seq2seq models, such as CNN seq2seq (Gehring et al., 2017) and transformer (Vaswani et al., 2017);
2. Tape-RNN, which show promising generalization results in various transduction tasks (Delétang et al., 2022);
3. and other novel transduction tasks.

Note: Task complexity is strongly tied to the structure of the learner. Thus, over-interpretations of our results beyond the context of this study (e.g., RNN seq2seq) are discouraged.

Acknowledgements

- The current research would not be initiated and successfully continued without the guidance and inspirations from **Jeffrey Heinz**.
- I am deeply grateful to the **three anonymous reviewers** for their constructive comments.
- I also thank **Jordan Kodner, William Oliver, Sarah Payne, Nicholas Behrje** who read through the early draft and provided helpful feedback.
- Parts of the work have been presented at various occasions at Stony Brook University, Yale University, University of Pennsylvania, and George Mason University as a talk or poster over the past few months, so my thanks also go for the audiences there.

Reproducibility

The source code, data, model training logs, trained models, and experimental results (raw or summarized) are open-sourced at `https://github.com/jaaack-wang/rnn-seq2seq-learning`.

References I

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994. doi: 10.1109/72.279181.

Mikolaj Bojanczyk, Sandra Kiefer, and Nathan Lhote. String-to-String Interpretations With Polynomial-Size Output. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 106:1–106:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-109-2. doi: 10.4230/LIPIcs.ICALP.2019.106. URL http://drops.dagstuhl.de/opus/volltexte/2019/10682.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. Massive exploration of neural machine translation architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1451, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1151. URL https://aclanthology.org/D17-1151.

Sarath Chandar, Chinnadhurai Sankar, Eugene Vorontsov, Samira Ebrahimi Kahou, and Yoshua Bengio. Towards non-saturating recurrent units for modelling long-term dependencies. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33013280. URL https://doi.org/10.1609/aaai.v33i01.33013280.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches, 2014. URL https://arxiv.org/abs/1409.1259.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. Neural networks and the chomsky hierarchy, 2022. URL https://arxiv.org/abs/2207.02098.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. ISSN 0364-0213. doi: https://doi.org/10.1016/0364-0213(90)90002-E. URL https://www.sciencedirect.com/science/article/pii/036402139090002E.

References II

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/gehring17a.html.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL https://aclanthology.org/D15-1166.

William Merrill. Sequential neural networks as automata. In *Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges*, pages 1–13, Florence, August 2019a. Association for Computational Linguistics. doi: 10.18653/v1/W19-3901. URL https://aclanthology.org/W19-3901.

William Merrill. Sequential neural networks as automata, 2019b. URL https://arxiv.org/abs/1906.01615.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/pascanu13.html.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

References III

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1 (2):270–280, 1989. doi: 10.1162/neco.1989.1.2.270.

Attention makes learning more efficient and robust

Follow-up experiment in total reduplication where attentional models only used 1/12 training examples, 1/9 parameter size, and 1/3 training epochs, compared to the attention-less ones.

	Attentional			Attention-less		
Dataset	**SRNN**	**GRU**	**LSTM**	**SRNN**	**GRU**	**LSTM**
Train	100.00	100.00	100.00	94.99	100.00	100.00
Test	99.20	99.53	99.58	84.93	90.21	91.86
Gen	35.20	14.07	19.37	0.00	5.10	4.54

GRU/LSTM seq2seq more expressive than SRNN seq2seq

With a consistent exception for reversal for unclear reasons.

		Attentional			Attention-less		
Task	**Dataset**	**SRNN**	**GRU**	**LSTM**	**SRNN**	**GRU**	**LSTM**
Identity	Train	100.00	100.00	100.00	69.74	98.26	100.00
	Test	99.97	**100.00**	**100.00**	42.82	70.46	**77.57**
	Gen	25.52	**37.41**	36.37	0.00	**10.41**	10.01
Rev	Train	100.00	100.00	100.00	100.00	100.00	100.00
	Test	**99.98**	99.87	99.88	**99.55**	88.46	92.85
	Gen	**40.14**	23.54	25.79	**23.89**	19.72	12.42
Total Red	Train	100.00	100.00	99.99	15.22	90.57	93.51
	Test	99.71	**99.77**	99.64	5.60	50.76	**55.17**
	Gen	**42.34**	23.23	20.31	0.00	4.39	**6.18**
Quad Copy	Train	2.43	79.84	82.73	1.62	49.29	67.29
	Test	1.99	67.75	**73.89**	0.61	27.76	**38.03**
	Gen	1.36	**8.20**	6.07	0.00	**0.85**	0.18
Average	Train	75.61	94.96	95.68	46.65	84.53	90.19
	Test	75.41	91.85	**93.35**	37.15	59.36	**65.91**
	Gen	**27.34**	23.10	22.13	5.97	**8.85**	7.20

GRU/LSTM seq2seq more expressive than SRNN seq2seq

GRU/LSTM seq2seq fits quadratic copying to certain extents, but SRNN seq2seq cannot. LSTM counts (Merrill, 2019a; Delétang et al., 2022).

		Attentional			Attention-less		
Task	**Dataset**	**SRNN**	**GRU**	**LSTM**	**SRNN**	**GRU**	**LSTM**
Identity	Train	100.00	100.00	100.00	69.74	98.26	100.00
	Test	99.97	**100.00**	**100.00**	42.82	70.46	**77.57**
	Gen	25.52	**37.41**	36.37	0.00	**10.41**	10.01
Rev	Train	100.00	100.00	100.00	100.00	100.00	100.00
	Test	**99.98**	99.87	99.88	**99.55**	88.46	92.85
	Gen	**40.14**	23.54	25.79	**23.89**	19.72	12.42
Total Red	Train	100.00	100.00	99.99	15.22	90.57	93.51
	Test	99.71	**99.77**	99.64	5.60	50.76	**55.17**
	Gen	**42.34**	23.23	20.31	0.00	4.39	**6.18**
Quad Copy	Train	2.43	79.84	82.73	1.62	49.29	67.29
	Test	1.99	67.75	**73.89**	0.61	27.76	**38.03**
	Gen	1.36	**8.20**	6.07	0.00	**0.85**	0.18
Average	Train	75.61	94.96	95.68	46.65	84.53	90.19
	Test	75.41	91.85	**93.35**	37.15	59.36	**65.91**
	Gen	**27.34**	23.10	22.13	5.97	**8.85**	7.20

SRNN seq2seq cannot count

Significantly enlarging model size for SRNN seq2seq helps little, if any: embedding size 128 $\rightarrow$ 384, hidden size 512 $\rightarrow$ 640/1024 (attn/attn-less).

	Attentional			Attention-less		
Dataset	**Full-seq**	**First *n*-symbol**	**Overlap**	**Full-seq**	**First *n*-symbol**	**Overlap**
Train	3.43	92.43	98.65	0.00	0.05	3.80
Test	3.00	90.92	98.53	0.00	0.05	3.81
Gen	2.79	84.23	92.82	0.00	0.19	3.68

SRNN seq2seq cannot count

SRNN seq2seq learns somehow periodically repeating the input sequences without knowing when to generate the end-of-sequence symbol!

	Test			Gen		
Model	**Run#1**	**Run#2**	**Run#3**	**Run#1**	**Run#2**	**Run#3**
SRNN	67.95	84.16	68.33	67.07	68.42	30.89
SRNN_{Large}	84.86	82.14	96.20	62.89	71.70	80.81
GRU	26.42	25.49	26.82	23.66	10.67	14.15
LSTM	26.83	25.51	25.52	6.07	8.72	7.56

The test/gen set first n-symbol accuracy (%) for all the attentional models trained for quadratic copying across three runs on the mapping $w \rightarrow w^{40}$. Full-sequence accuracy always is 0.00%, since the mapping is not what the models were trained for.