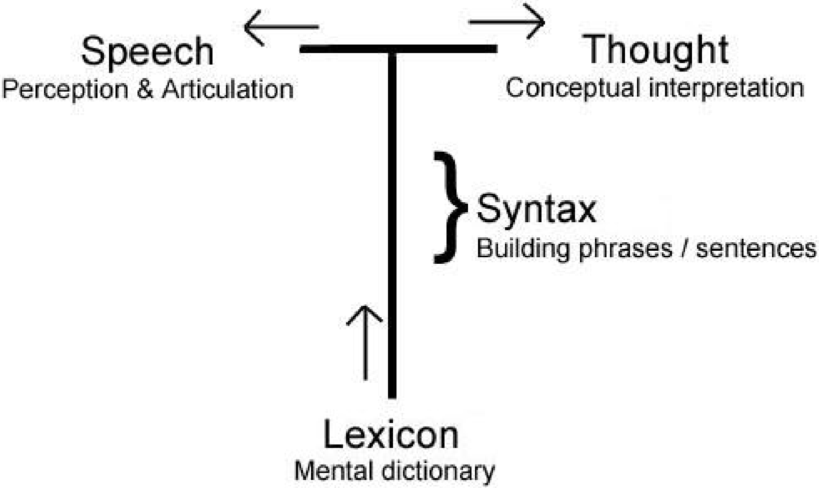
Gaël Le Godais[1,2], Tal Linzen[1,3]
and Emmanuel Dupoux[1]

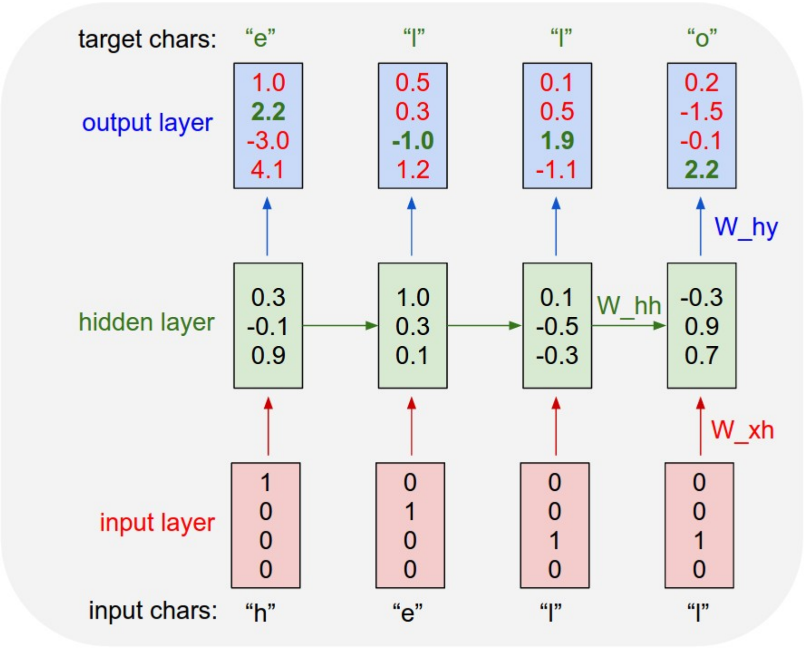
[1] LSCP & IJN, ENS Paris
[2] ENSIMAG
[3] Johns Hopkins University

Comparing Character-level Neural Language Models Using a Lexical Decision Task


Speech
Perception & Articulation
Thought
Conceptual interpretation
Syntax
Building phrases / sentences
Lexicon
Mental dictionary

Character-level Convolutional Networks for Text Classification*

Xiang Zhang **Junbo Zhao** **Yann LeCun**
Courant Institute of Mathematical Sciences, New York University
719 Broadway, 12th Floor, New York, NY 10003
{xiang, junbo.zhao, yann}@cs.nyu.edu

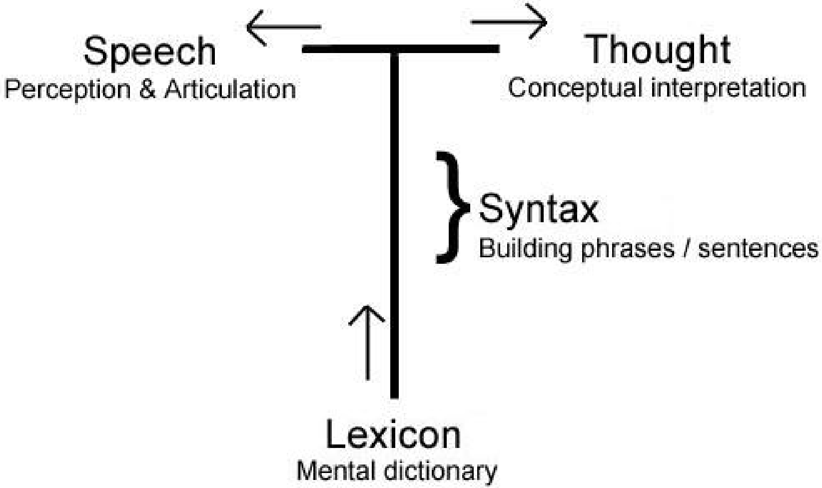

target chars:
“e”
“l”
“l”
“o”
output layer
1.0
2.2
-3.0
4.1
0.5
0.3
-1.0
1.2
0.1
0.5
1.9
-1.1
0.2
-1.5
-0.1
2.2
W_hy
hidden layer
0.3
-0.1
0.9
1.0
0.3
0.1
0.1
-0.5
-0.3
W_hh
-0.3
0.9
0.7
W_xh
input layer
1
0
0
0
0
1
0
0
0
0
1
0
0
0
1
0
input chars:
“h”
“e”
“l”
“l”

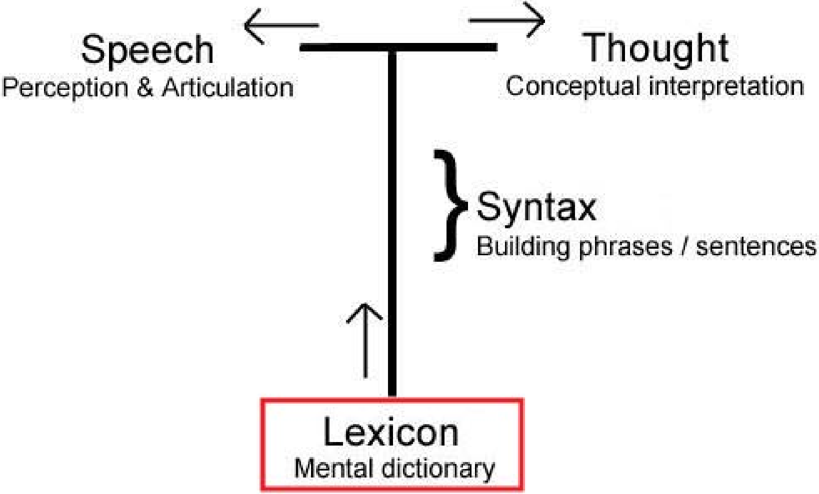
PANDARUS:
Alas, I think he shall be come approached and the day When little srain would be attain'd into being never fed, And who is but a chain and subjects of his death, I should not sleep.

(Karpathy et al., 2016)

PANDARUS:
Alas, I think he shall be come approached and the day When little srain would be attain'd into being never fed, And who is but a chain and subjects of his death, I should not sleep.

(Karpathy et al., 2016)


Speech
Perception & Articulation
Thought
Conceptual interpretation
Syntax
Building phrases / sentences
Lexicon
Mental dictionary


Speech
Perception & Articulation
Thought
Conceptual interpretation
Syntax
Building phrases / sentences
Lexicon
Mental dictionary

Contributions

Contributions

- We introduce a psycholinguistic task that evaluates the lexicon implicit in a character-level language model

Contributions

- We introduce a psycholinguistic task that evaluates the lexicon implicit in a character-level language model
- As a first use of the task, we evaluate how the size and depth of the network affect its lexical capacity

Lexical decision

(Rubenstein et al., 1970)

Lexical decision
(Rubenstein et al., 1970)

+

Lexical decision
(Rubenstein et al., 1970)

plurb

Lexical decision
(Rubenstein et al., 1970)

NONWORD

Lexical decision
(Rubenstein et al., 1970)

+

Lexical decision
(Rubenstein et al., 1970)

bowl

Lexical decision
(Rubenstein et al., 1970)

WORD

Lexical decision using a language model

Lexical decision using a language model

$$P(\text{bowl}) = P(\text{b}) + P(\text{o}|\text{b}) + P(\text{w}|\text{bo}) + P(\text{l}|\text{bow})$$

Lexical decision using a language model

$$P(\text{bowl}) = P(\text{b}) + P(\text{o}|\text{b}) + P(\text{w}|\text{bo}) + P(\text{l}|\text{bow})$$

Words “should” have a higher probability than nonwords...

$$P(\text{bowl}) > \epsilon$$

Lexical decision using a language model

$$P(\text{bowl}) = P(\text{b}) + P(\text{o}|\text{b}) + P(\text{w}|\text{bo}) + P(\text{l}|\text{bow})$$

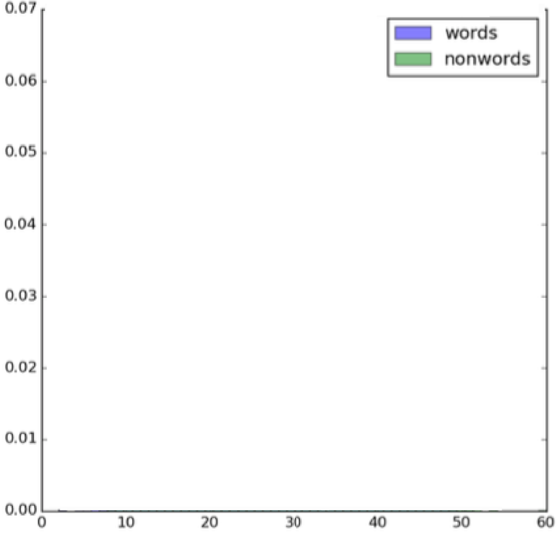
Words “should” have a higher probability than nonwords...

$$P(\text{bowl}) > \epsilon$$

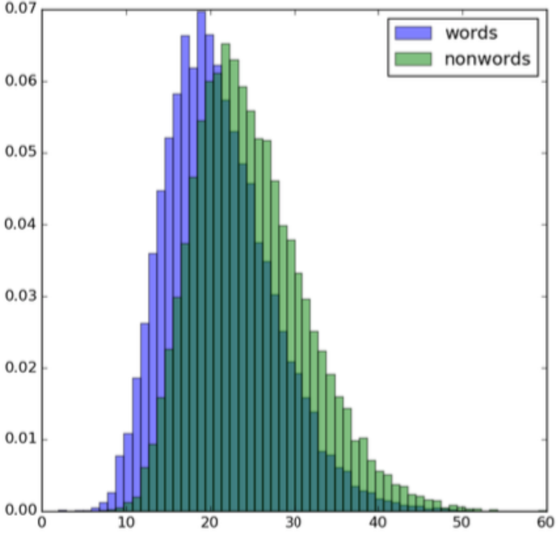
$$P(\text{plurb}) < \epsilon$$

(Lau et al., 2016)

LM negative log-probabilities
(Balota et al., 2007)



LM negative log-probabilities
(Balota et al., 2007)



Issues with a fixed probability threshold

- Every character has probability $< 1 \quad \rightarrow$
  long words may have lower probability than short nonwords

Issues with a fixed probability threshold

- Every character has probability $< 1 \quad \rightarrow$
  long words may have lower probability than short nonwords
- Rare characters (*q*, *z*) have lower probability than frequent ones (*b*)

Issues with a fixed probability threshold

- Every character has probability $< 1 \quad \rightarrow$
  long words may have lower probability than short nonwords
- Rare characters (*q*, *z*) have lower probability than frequent ones (*b*)
- Rare transitions (*zk*) have lower probability than frequent ones (*ba*)

Issues with a fixed probability threshold

- Every character has probability $< 1 \quad \rightarrow$ long words may have lower probability than short nonwords
- Rare characters (*q*, *z*) have lower probability than frequent ones (*b*)
- Rare transitions (*zk*) have lower probability than frequent ones (*ba*)
- Can try to normalize for these factors (Berg-Kirkpatrick et al., 2012; Lau et al., 2016)

Issues with a fixed probability threshold

- Every character has probability $< 1 \quad \rightarrow$
  long words may have lower probability than short nonwords
- Rare characters (*q*, *z*) have lower probability than frequent ones (*b*)
- Rare transitions (*zk*) have lower probability than frequent ones (*ba*)
- Can try to normalize for these factors (Berg-Kirkpatrick et al., 2012; Lau et al., 2016)
- Alternative: matching (Linzen et al., 2016)

Spot-the-word (2AFC lexical decision)
(Baddeley et al., 1993)

Spot-the-word (2AFC lexical decision)
(Baddeley et al., 1993)

bowl vowl

Spot-the-word (2AFC lexical decision)
(Baddeley et al., 1993)

bowl   vowl

Spot-the-word (2AFC lexical decision)
(Baddeley et al., 1993)

+

Spot-the-word (2AFC lexical decision)
(Baddeley et al., 1993)

poat moat

Spot-the-word (2AFC lexical decision)

(Baddeley et al., 1993)

poat moat

Spot-the-word (2AFC lexical decision)

(Baddeley et al., 1993)

+

Spot-the-word (2AFC lexical decision)
(Baddeley et al., 1993)

enacity emacity

Spot-the-word (2AFC lexical decision)
(Baddeley et al., 1993)

enacity emacity

Performing the task

$$P(\text{'bat'}) > P(\text{'bap'})?$$

Performing the task

$P(\text{'bat'}) > P(\text{'bap'})$?

battery, wombat, debate

Performing the task

$$P(\text{'bat'}) > P(\text{'bap'})?$$

battery, wombat, debate

Performing the task

$$P(\text{'bat'}) > P(\text{'bap'})?$$

battery, wombat, debate

$$P(\text{' bat '}) > P(\text{' bap '})?$$

Contributions

- We introduce a psycholinguistic task that evaluates the lexicon implicit in a character-level language model
- As a first use of the task, we evaluate how the size and depth of the network affect its lexical capacity

Experimental design

Experimental design

- Architecture: SRN vs. LSTM

Experimental design

- Architecture: SRN vs. LSTM
- Number of layers: 1, 2 or 3

Experimental design

- Architecture: SRN vs. LSTM
- Number of layers: 1, 2 or 3
- Number of units per layer: 16, 32, 64 or 128

Experimental design

- Architecture: SRN vs. LSTM
- Number of layers: 1, 2 or 3
- Number of units per layer: 16, 32, 64 or 128
- Six random seeds for each combination

Experimental design

- Architecture: SRN vs. LSTM
- Number of layers: 1, 2 or 3
- Number of units per layer: 16, 32, 64 or 128
- Six random seeds for each combination
- Trained on 10M words (50M characters) from the movie/book corpus (Zhu et al., 2015)

Nonword generation

- Nonwords matched for length and bigram probability (respecting position and syllable structure) using Wuggy (Keuleers & Brysbaert, 2010):

Nonword generation

- Nonwords matched for length and bigram probability (respecting position and syllable structure) using Wuggy (Keuleers & Brysbaert, 2010):

travel	chavel

Nonword generation

- Nonwords matched for length and bigram probability (respecting position and syllable structure) using Wuggy (Keuleers & Brysbaert, 2010):

travel	chavel
assimilated	assitilated

Nonword generation

- Nonwords matched for length and bigram probability (respecting position and syllable structure) using Wuggy (Keuleers & Brysbaert, 2010):

travel	chavel
assimilated	assitilated
copious	conious

Nonword generation

- Nonwords matched for length and bigram probability (respecting position and syllable structure) using Wuggy (Keuleers & Brysbaert, 2010):

travel	chavel
assimilated	assitilated
copious	conious
fib	wib

Nonword generation

- Nonwords matched for length and bigram probability (respecting position and syllable structure) using Wuggy (Keuleers & Brysbaert, 2010):

travel	chavel
assimilated	assitilated
copious	conious
fib	wib
needed	nooded

Baselines

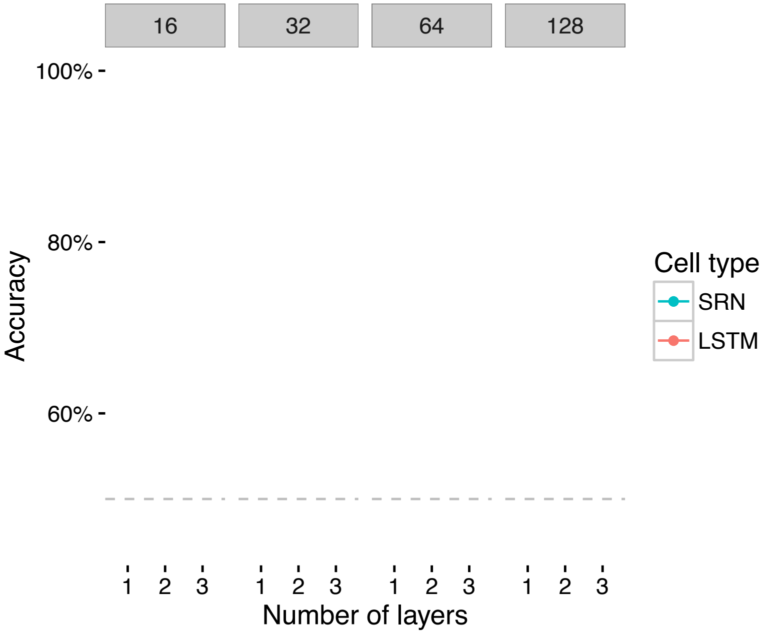
- Chance: 50% accuracy

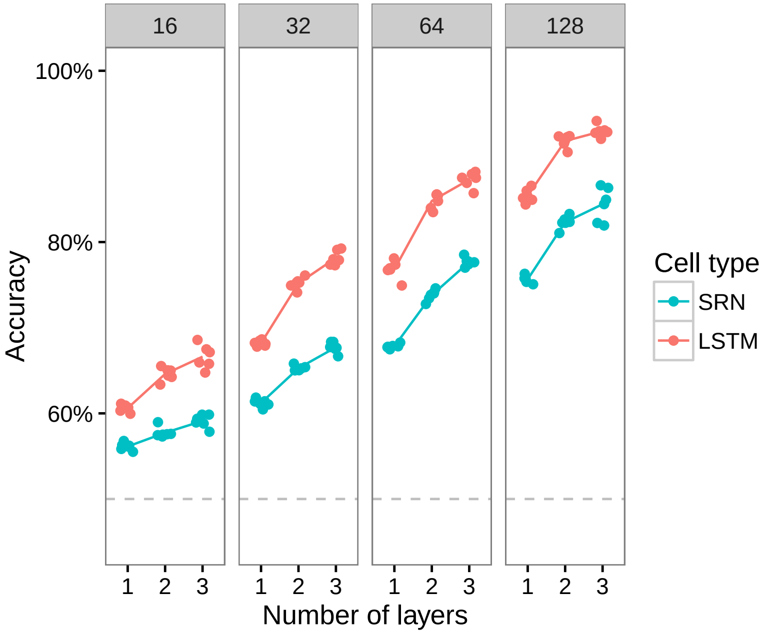
Baselines

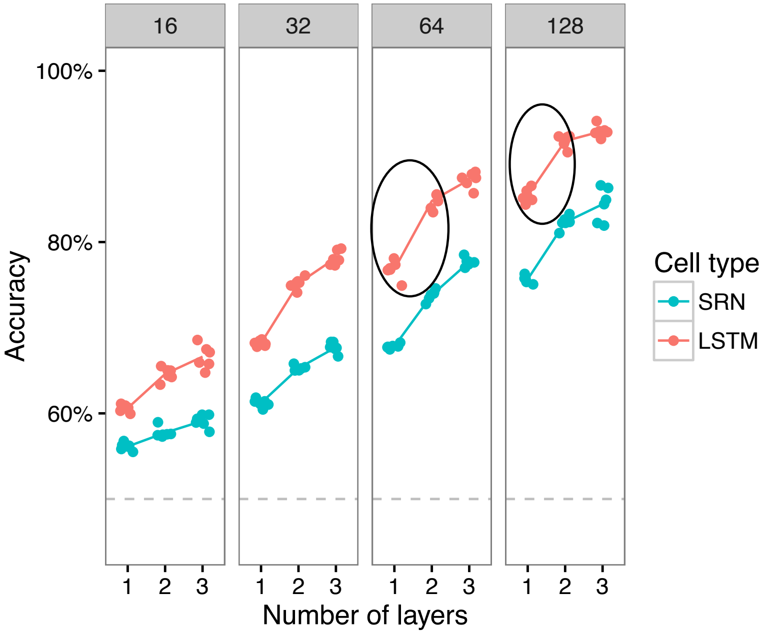
- Chance: 50% accuracy
- Character unigram language model: 49.6% accuracy

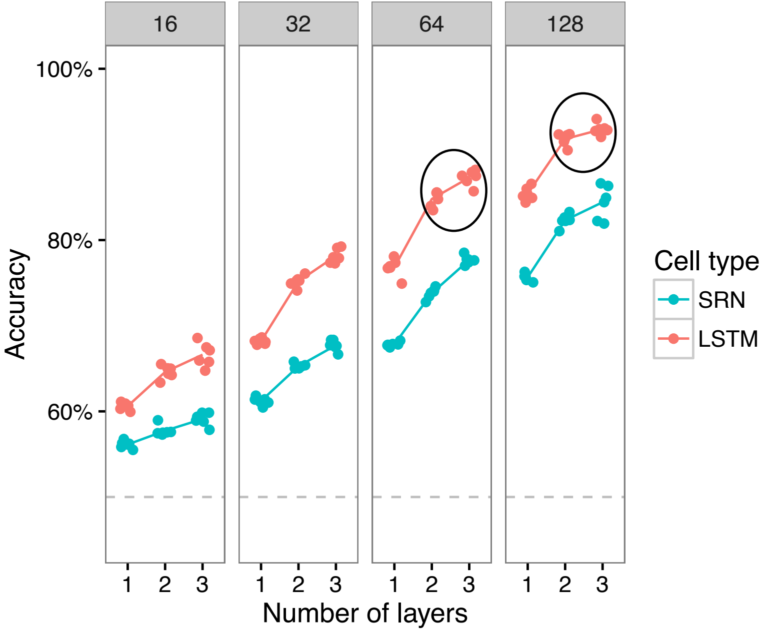
Baselines

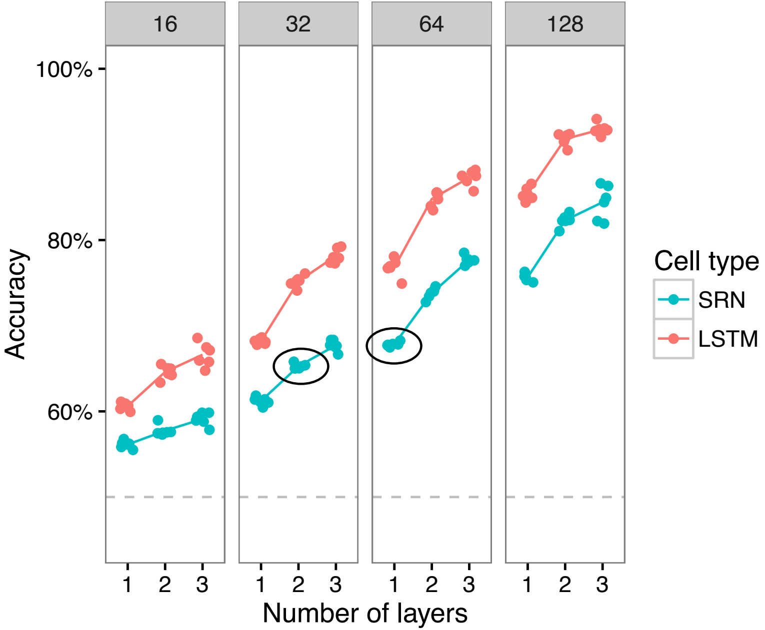
- Chance: 50% accuracy
- Character unigram language model: 49.6% accuracy
- Character bigram language model: 52.1% accuracy

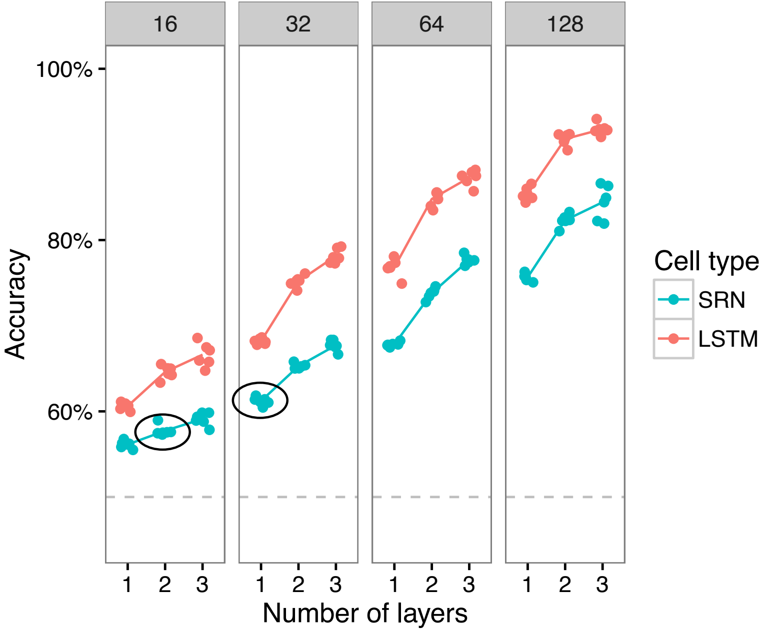

16
32
64
128
100%
80%
60%
Accuracy
1 2 3 1 2 3 1 2 3 1 2 3
Number of layers
Cell type
SRN
LSTM


16
32
64
128
100%
80%
60%
Accuracy
1
2
3
Number of layers
Cell type
SRN
LSTM


16
32
64
128
100%
80%
60%
Accuracy
1 2 3
Number of layers
Cell type
SRN
LSTM


16
32
64
128
100%
80%
60%
Accuracy
1
2
3
Number of layers
Cell type
SRN
LSTM


16
32
64
128
100%
80%
60%
Accuracy
1
2
3
Number of layers
Cell type
SRN
LSTM


16
32
64
128
100%
80%
60%
Accuracy
1
2
3
Number of layers
Cell type
SRN
LSTM

Number of parameters

Number of parameters

SRN:

$$\mathbf{s_i} = (\mathbf{x_i}\mathbf{W^x} + \mathbf{s_{i-1}}\mathbf{W^s} + \mathbf{b})$$

Number of parameters

SRN:

$$\mathbf{s_i} = (\mathbf{x_i}\mathbf{W^x} + \mathbf{s_{i-1}}\mathbf{W^s} + \mathbf{b})$$

LSTM:

$$
\begin{aligned}
\mathbf{c_j} &= \mathbf{c_{j-1}} \odot \mathbf{f} + \mathbf{g} \odot \mathbf{i} \\
\mathbf{h_j} &= \tanh(\mathbf{c_j}) \odot \mathbf{o} \\
\mathbf{i} &= \sigma(\mathbf{x_j}\mathbf{W^{xi}} + \mathbf{h_{j-1}}\mathbf{W^{hi}}) \\
\mathbf{f} &= \sigma(\mathbf{x_j}\mathbf{W^{xf}} + \mathbf{h_{j-1}}\mathbf{W^{hf}}) \\
\mathbf{o} &= \sigma(\mathbf{x_j}\mathbf{W^{xo}} + \mathbf{h_{j-1}}\mathbf{W^{ho}}) \\
\mathbf{g} &= \tanh(\mathbf{x_j}\mathbf{W^{xg}} + \mathbf{h_{j-1}}\mathbf{W^{hg}})
\end{aligned}
$$

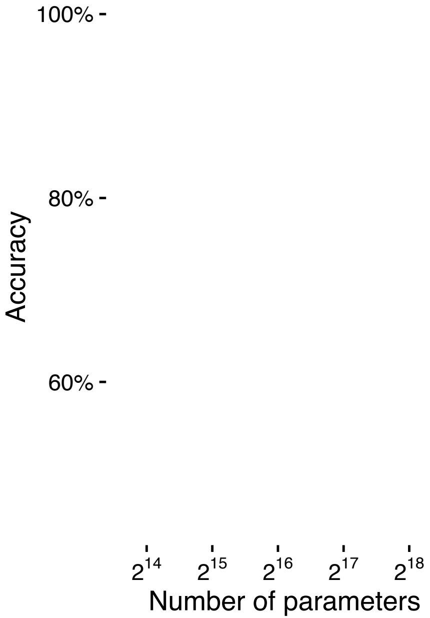
Number of parameters

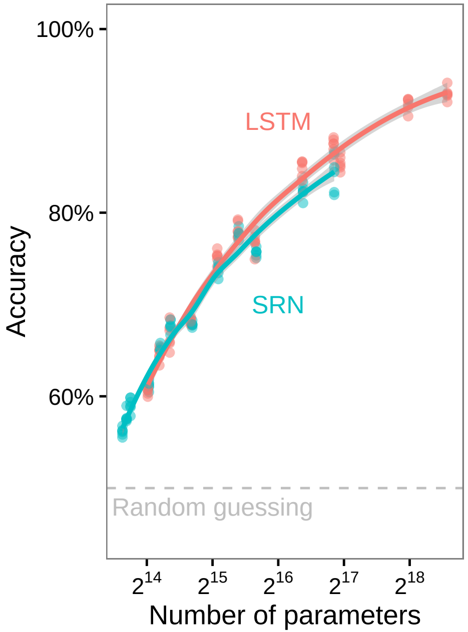
SRN:

$$\mathbf{s_i} = (\mathbf{x_i}\mathbf{W^x} + \mathbf{s_{i-1}}\mathbf{W^s} + \mathbf{b})$$

LSTM:

$$\begin{aligned}
\mathbf{c_j} &= \mathbf{c_{j-1}} \odot \mathbf{f} + \mathbf{g} \odot \mathbf{i} \\
\mathbf{h_j} &= \tanh(\mathbf{c_j}) \odot \mathbf{o} \\
\mathbf{i} &= \sigma(\mathbf{x_j}\mathbf{W^{xi}} + \mathbf{h_{j-1}}\mathbf{W^{hi}}) \\
\mathbf{f} &= \sigma(\mathbf{x_j}\mathbf{W^{xf}} + \mathbf{h_{j-1}}\mathbf{W^{hf}}) \\
\mathbf{o} &= \sigma(\mathbf{x_j}\mathbf{W^{xo}} + \mathbf{h_{j-1}}\mathbf{W^{ho}}) \\
\mathbf{g} &= \tanh(\mathbf{x_j}\mathbf{W^{xg}} + \mathbf{h_{j-1}}\mathbf{W^{hg}})
\end{aligned}$$


100%
80%
60%
Accuracy
2^{14} 2^{15} 2^{16} 2^{17} 2^{18}
Number of parameters



Accuracy
100%
80%
60%
LSTM
SRN
Random guessing
2^14
2^15
2^16
2^17
2^18
Number of parameters

Future work

- Memorization vs. generalization: what does the network learn about productive word-formation rules (*neural* $\rightarrow$ *neuralize*)?

Future work

- Memorization vs. generalization: what does the network learn about productive word-formation rules (*neural* $\rightarrow$ *neuralize*)?
    - Preliminary results: a lot

Future work

- Memorization vs. generalization: what does the network learn about productive word-formation rules (*neural* $\rightarrow$ *neuralize*)?
  - Preliminary results: a lot
- Frequency effects

Future work

- Memorization vs. generalization: what does the network learn about productive word-formation rules (*neural* $\rightarrow$ *neuralize*)?
  - Preliminary results: a lot
- Frequency effects
  - Preliminary results: there are

Future work

- Memorization vs. generalization: what does the network learn about productive word-formation rules (*neural* $\rightarrow$ *neuralize*)?
  - Preliminary results: a lot
- Frequency effects
  - Preliminary results: there are
  - Lexical decision accuracy and perplexity will diverge the most on rare words

Future work

- Memorization vs. generalization: what does the network learn about productive word-formation rules (*neural* $\rightarrow$ *neuralize*)?
  - Preliminary results: a lot
- Frequency effects
  - Preliminary results: there are
  - Lexical decision accuracy and perplexity will diverge the most on rare words
- To compare to humans datasets (Balota et al., 2007), we need more research on doing the yes/no version of the task

Future work

- Memorization vs. generalization: what does the network learn about productive word-formation rules (*neural* $\rightarrow$ *neuralize*)?
  - Preliminary results: a lot
- Frequency effects
  - Preliminary results: there are
  - Lexical decision accuracy and perplexity will diverge the most on rare words
- To compare to humans datasets (Balota et al., 2007), we need more research on doing the yes/no version of the task
  - Map language model probabilities to a lexicality judgment

Future work

- Memorization vs. generalization: what does the network learn about productive word-formation rules (*neural* $\rightarrow$ *neuralize*)?
  - Preliminary results: a lot
- Frequency effects
  - Preliminary results: there are
  - Lexical decision accuracy and perplexity will diverge the most on rare words
- To compare to humans datasets (Balota et al., 2007), we need more research on doing the yes/no version of the task
  - Map language model probabilities to a lexicality judgment
  - Train classifier to predict lexicality from the state of the RNN

Future work

- Memorization vs. generalization: what does the network learn about productive word-formation rules (*neural* $\rightarrow$ *neuralize*)?
  - Preliminary results: a lot
- Frequency effects
  - Preliminary results: there are
  - Lexical decision accuracy and perplexity will diverge the most on rare words
- To compare to humans datasets (Balota et al., 2007), we need more research on doing the yes/no version of the task
  - Map language model probabilities to a lexicality judgment
  - Train classifier to predict lexicality from the state of the RNN
  - Run humans on the spot-the-word task

Conclusions

- We have proposed a tool for studying lexical learning in character-level RNNs

Conclusions

- We have proposed a tool for studying lexical learning in character-level RNNs
- Our first study using this tool showed that large enough networks can perform the task well, despite not being trained on it

Conclusions

- We have proposed a tool for studying lexical learning in character-level RNNs
- Our first study using this tool showed that large enough networks can perform the task well, despite not being trained on it
- The number of parameters is by far the most important determinant of performance: depth isn't useful

Conclusions

- **The human mind is at least as much of a black box as neural networks**

Conclusions

- **The human mind is at least as much of a black box as neural networks**
- Tasks from psycholinguistics can be used to better understand neural networks

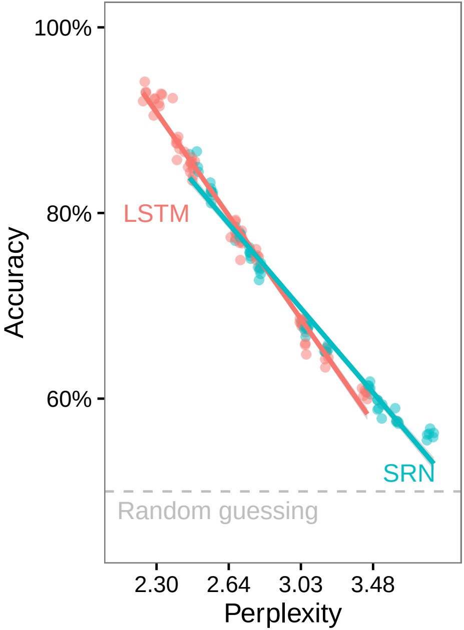
Conclusions

- **The human mind is at least as much of a black box as neural networks**
- Tasks from psycholinguistics can be used to better understand neural networks
- Our code is available at `https://github.com/bootphon/char_rnn_lexical_decision`

Acknowledgements

- European Research Council (grant ERC-2011-AdG 295810 BOOTPHON)
- Agence Nationale pour la Recherche (grants ANR-10-IDEX-0001-02 PSL and ANR-10-LABX-0087 IEC)

Thank you!


100%
80%
60%
Accuracy
LSTM
SRN
Random guessing
2.30
2.64
3.03
3.48
Perplexity

Balota, D. A., Yap, M. J., Cortese, M. J., Hutchison, K. A., Kessler, B., Loftis, B., … Treiman, R. (2007). The English lexicon project. *Behavior Research Methods*, *39*(3), 445–459.

Berg-Kirkpatrick, T., Burkett, D., & Klein, D. (2012). An empirical investigation of statistical significance in nlp. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 995–1005).

Karpathy, A., Johnson, J., & Li, F.-F. (2016). Visualizing and understanding recurrent networks. In *Proceedings of international conference on learning representations.*

Keuleers, E., & Brysbaert, M. (2010). Wuggy: A multilingual pseudoword generator. *Behavior Research Methods*, *42*(3), 627–633. Retrieved from `http://dx.doi.org/10.3758/BRM.42.3.627` doi: 10.3758/BRM.42.3.627

Lau, J. H., Clark, A., & Lappin, S. (2016). Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*.

Linzen, T., Dupoux, E., & Goldberg, Y. (2016). Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, *4*, 521–535.