Long-Short Range Context Neural Networks for Language Modeling

Youssef Oualil Mittul Singh Clayton Greenberg Dietrich Klakow

firstName.lastName@lsv.uni-saarland.de

Spoken Language Systems, Saarland University, Saarland Informatics Campus, Saarbrücken, Germany

Short vs Long Range Dependencies in LM

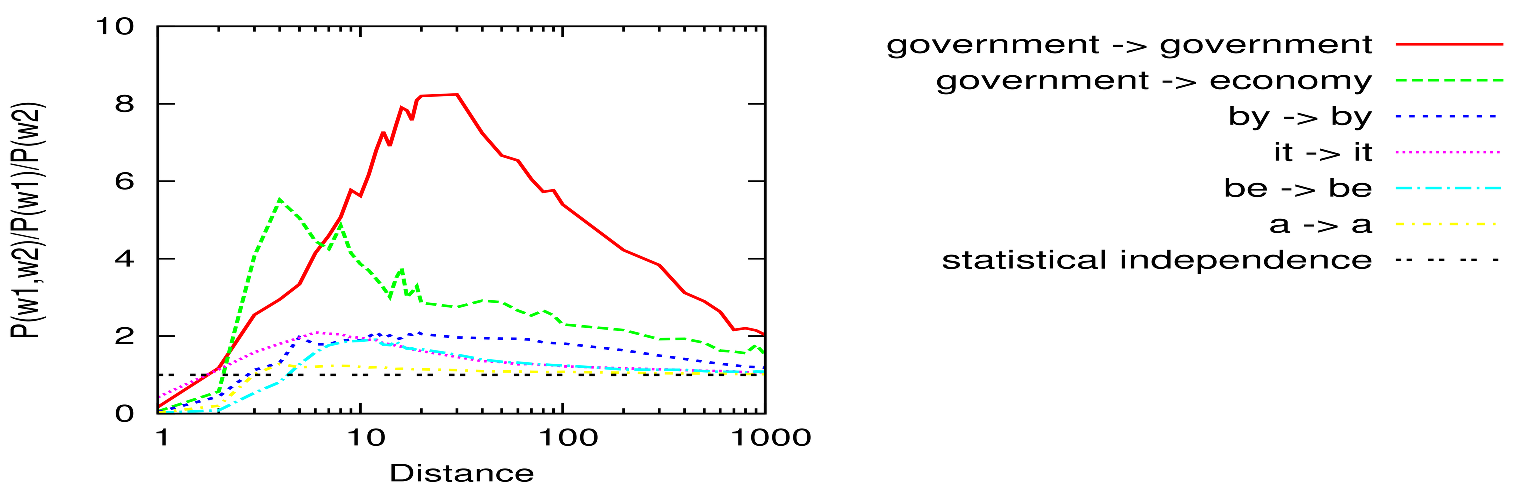
Language models (LMs) predict upcoming text/speech.
Traditionally: use the most recent n words (n-gram LM):

$$p(\textcolor{red}{\text{upcoming word}} \mid \textcolor{blue}{\text{past words}}) \approx p(\textcolor{red}{\text{upcoming word}} \mid \textcolor{blue}{\text{last } n \text{ words}})$$

For $n > 5$, n-gram LMs often fail: too sparse and complex.
But how much would long distance relationships help?

Measuring correlations in language, at a distance d:

$$c_d(w_1, w_2) = \frac{p_d(w_1, w_2)}{p(w_1)\cdot p(w_2)}$$



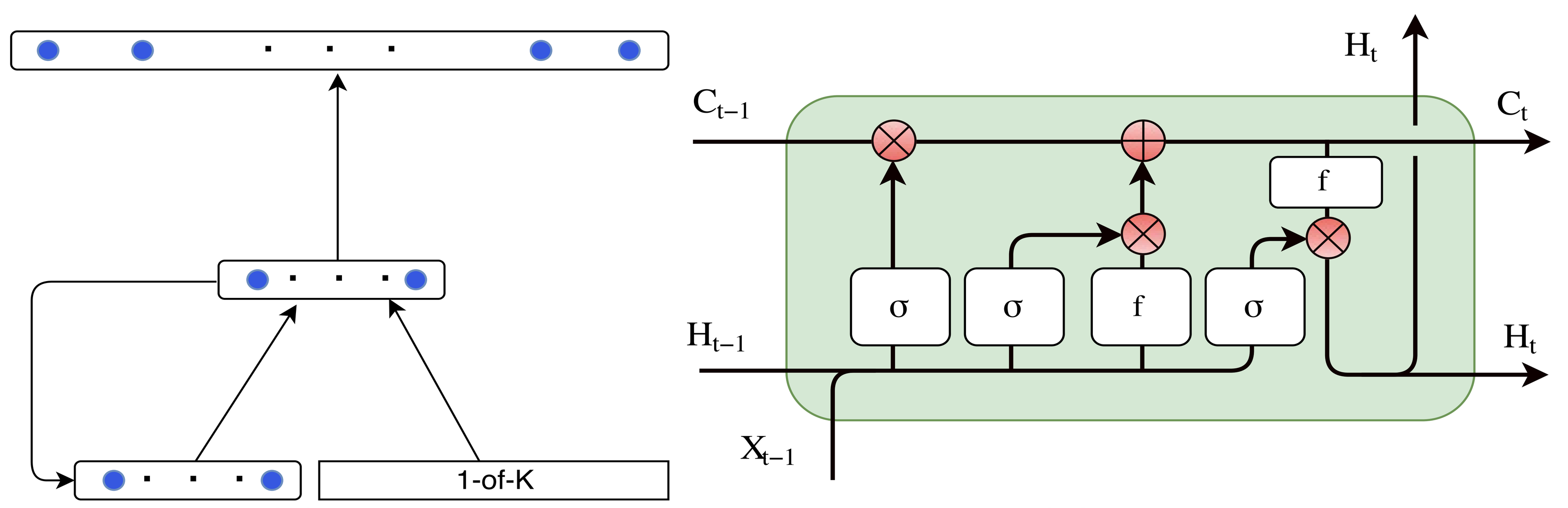
→ predict the next word based on the current word *and* a hidden memory state that evolves over time.

Recurrent Networks for Language Modeling

Recurrent Language models use a dynamic hidden state to model the context:

$$p(\textcolor{red}{\text{upcoming word}} \mid \textcolor{blue}{\text{past words}}) \approx p(\textcolor{red}{\text{upcoming word}} \mid \textcolor{blue}{\text{context state}})$$

Different recurrent neural network models can be used to dynamically update the hidden context state.



(a) RNN (b) LSTM

Figure : *RNN vs LSTM architectures.*

Multi-Span Language Models

Observations

1) RNN hidden state changes rapidly → short context.
2) LSTM does not explicitly model short vs long context.

Solution

Use a multi-span network with two memory states to explicitly and separately model the short and long range dependencies.

Long-Short Range Context Neural Networks

LSRC network takes advantage of the LSTM ability to model long range context while, simultaneously, learning and integrating the short context through an additional recurrent local state.

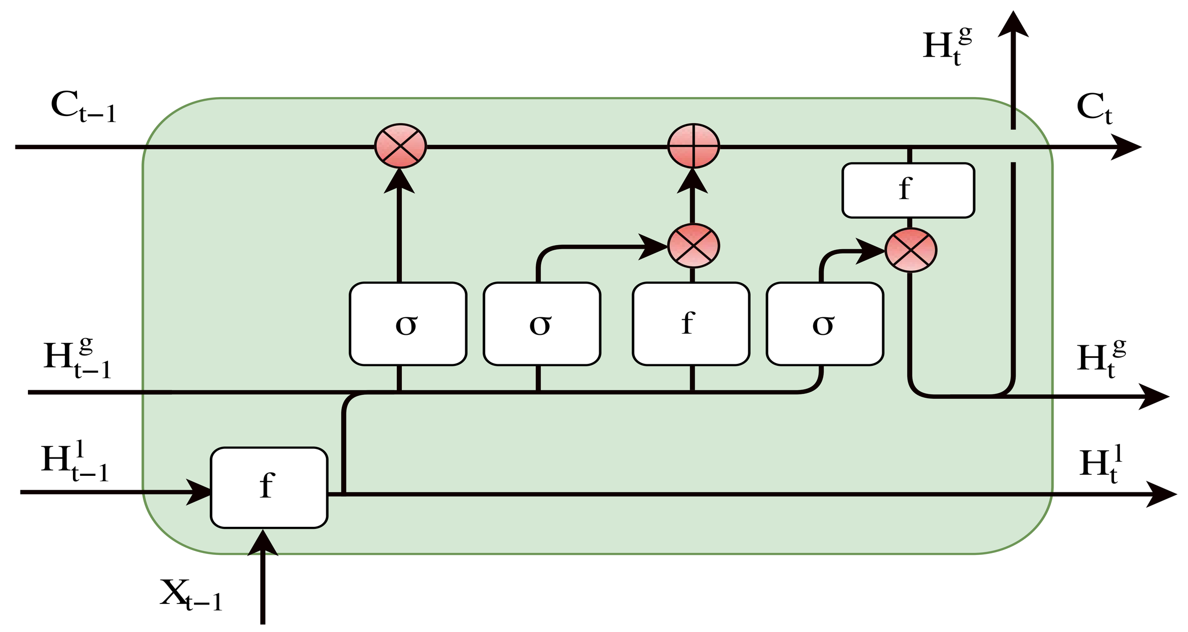


Figure : Block diagram of the recurrent module of an LSRC network.

$$H_t^l = f\left(X_{t-1} + U_l^c \cdot H_{t-1}^l\right)$$
$$\{i,f,o\}_t = \sigma\left(V_l^{i,f,o} \cdot H_t^l + V_g^{i,f,o} \cdot H_{t-1}^g\right)$$
$$\tilde{C}_t = f\left(V_l^c \cdot H_t^l + V_g^c \cdot H_{t-1}^g\right)$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$
$$H_t^g = o_t \odot f\left(C_t\right)$$
$$P_t = g\left(W \cdot H_t^g\right)$$

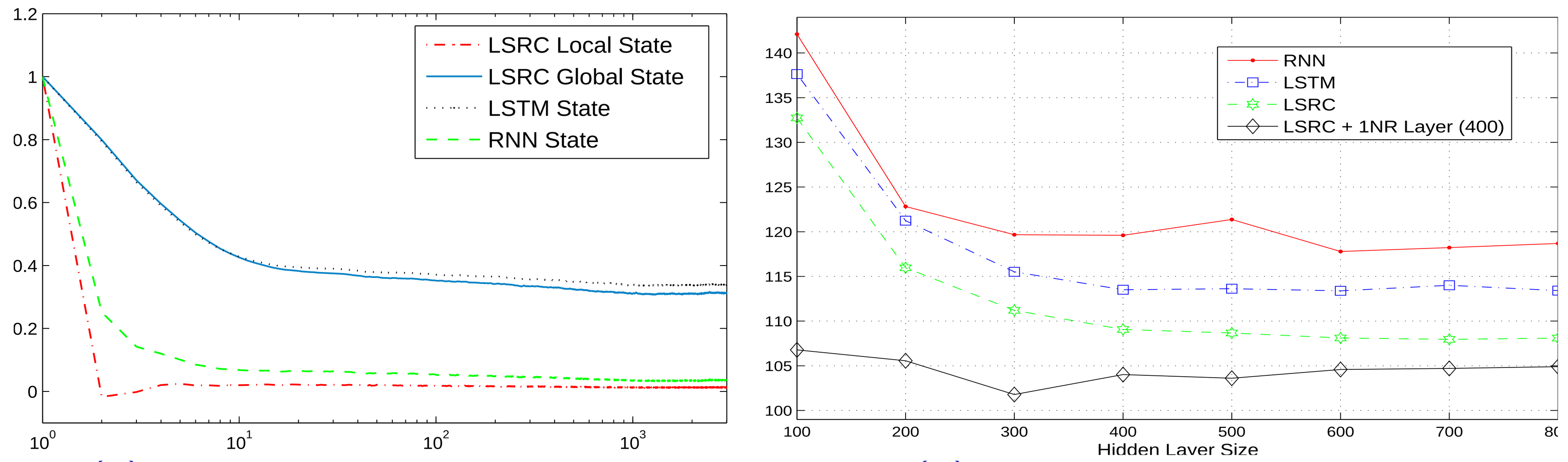
LSRC Properties

1) Captures RNN and LSTM properties in a single network.
2) Uses two separate hidden memory states.
3) Explicitly and separately models short (local) vs long (global) context.
4) Recursively updates the global context using local context.

Temporal Correlation and Perplexity

LM experiments conducted on the PTB and LTCB corpus:

Corpus	Train	Dev	Test
PTB	930K	74K	82K
LTCB	133M	7.8M	7.9M



(a) Temporal correlation on PTB corpus (b) Perplexity on PTB corpus

	model PPL			model+KN5 PPL			# of Par.
N-1=	1	2	4	1	2	4	4
KN+cache	168	134	129	—	—	—	—
FFNN	176	131	119	132	116	107	6.32M
RNN		117			104		8.16M
LSTM (1L)		113			99		6.96M
LSRC(100)		109			96		5.81M
LSRC(200)		104			94		7.0M

Table : LMs performance on the PTB test set.

	model PPL			# of Par.
Context Size M=N-1	1	2	4	4
KN+cache	188	127	109	—
FFNN [M*200]-600-600-80k	235	150	114	64.84M
RNN [600]-R600-80k		85		96.36M
LSTM [200]-R600-80k		66		65.92M
LSTM [200]-R600-R600-80k		61		68.80M
LSRC [200]-R600-80k		63		65.96M
LSRC [200]-R600-600-80k		59		66.32M

Table : LMs performance on the LTCB test set.

Conclusion

LSRC outperforms state-of-the-art NNLMs by explicitly modeling long vs short range context using two separate (local and global) memory states.

Conference on Empirical Methods in Natural Language Processing (EMNLP), 1-5 November 2016, Austin, Texas, USA.