Exploring the Limits of Language Modeling

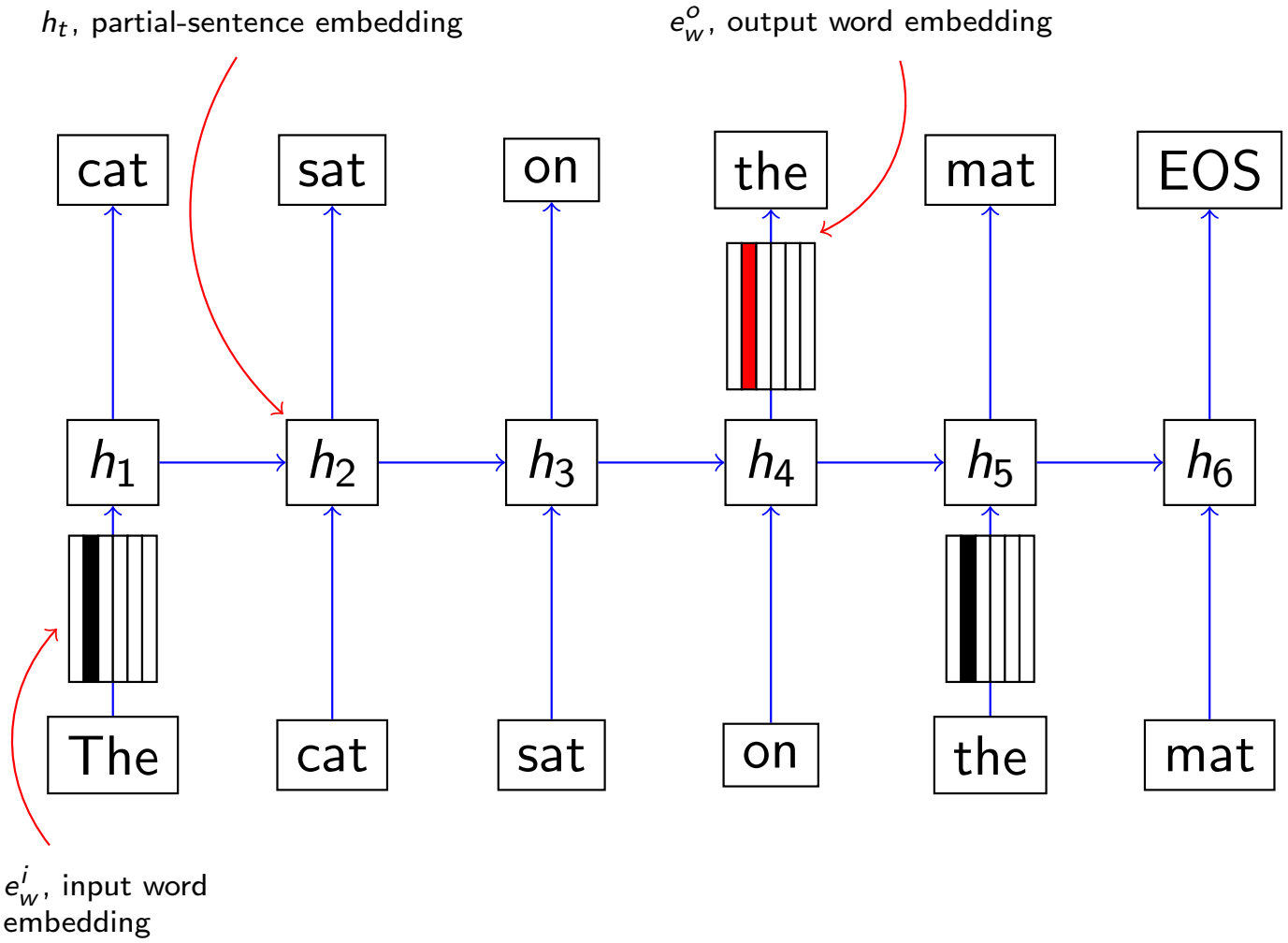
Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, Yonghui Wu

Presented by Arvid Frydenlund

November 11, 2016

Word-level Neural Language Modelling

$$p(w) = \frac{\exp(z_w)}{\sum_{w' \in V} \exp(z'_w)} \text{ where } z_w = h_t^T e_w^o$$



Overview

They present 4 different models:

1. Word-level language model
2. Character-level input word-level output, without an input look-up table
3. Character-level input word-level output, without a any look-up table
4. Word-level input, character-level output, with a encoder-decoder

Models

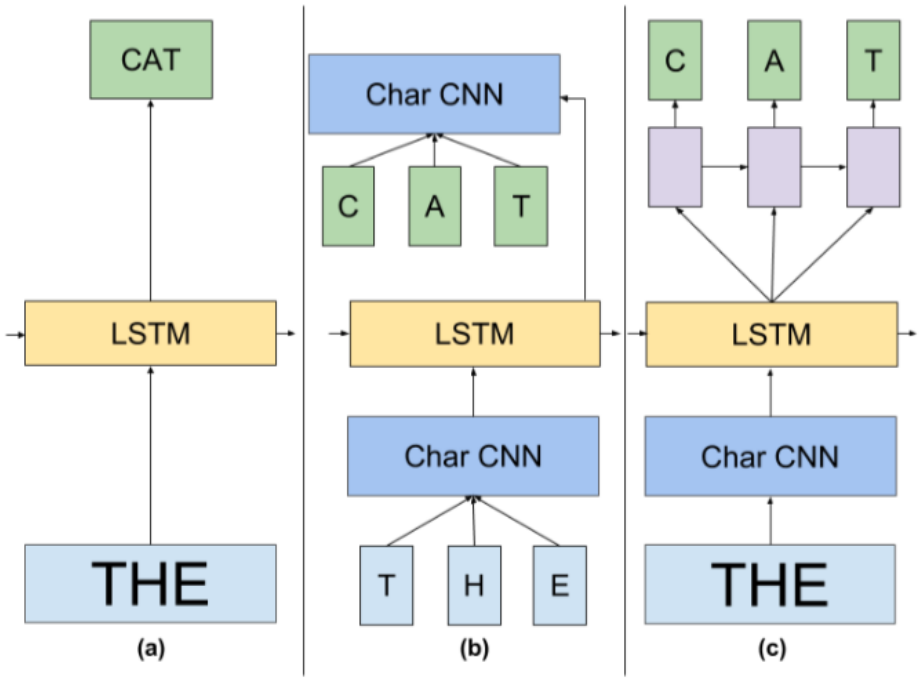


Figure 1. A high-level diagram of the models presented in this paper. (a) is a standard LSTM LM. (b) represents an LM where both input and Softmax embeddings have been replaced by a character CNN. In (c) we replace the Softmax by a next character prediction LSTM network.

Achievements:

- State-of-the-art language modelling on Billion Word Benchmark (800k vocabulary)
- Reduced perplexity from 51.3 to 30.0, and then to 23.7 with an ensemble
- While significantly reducing model parameters (20 billion to 1.04 billion)
- Novel replacement of the output look-up table
- Novel encoder-decoder model for character-level ouput language modelling

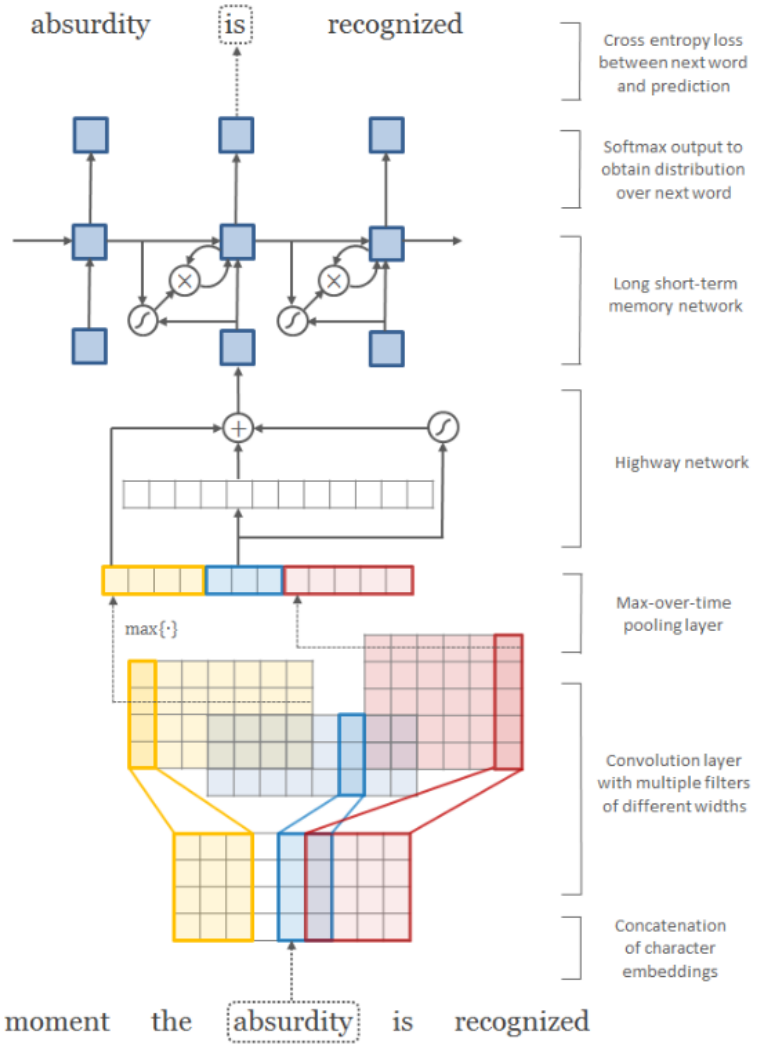
Issues:

- Full softmax over 800k vocabulary at test time
- Training time (32 GPUs for 3 weeks)
- Output look-up table replacement preforms worse than a full look-up table and still requires one anyways
- Character-level output model doesn't work well
- Issue of comparing character-level output to word-level output

Modelling input words

- Don't model words independently
- 'cat' and 'cats' should share semantic information
- '-ing' should share syntactic information
- Replace whole word look-up table with compositional function
- c,a,t,s $\rightarrow e_w^i$
- Can be seen as approximating the look-up table with an embedded neural network.
  - **Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation**, Ling *et al.* 2015
  - used bidirectional LSTM
  - **Character-Aware Neural Language Models**, Kim *et al.*, 2015
  - used a CNN & Highway feedforward NN

Char-CNN (Kim *et al.*)



Results of replacing input look-up table

Model (RNN state size, e^o_w size)	Test Perplexity	Params (B)
Previous SOTA	51.3	20
LSTM (512, 512)	54.1	0.82
LSTM (1024, 512)	48.2	0.82
LSTM (2048, 512)	43.7	0.83
LSTM (8192, 2048), No dropout	37.9	3.3
LSTM (8192, 2048), Dropout	32.2	3.3
2-layer LSTM (8192, 1024), Big LSTM	30.6	1.8
Big LSTM with CNN Inputs	**30.0**	**1.04**

Modelling output words

- c,a,t,s $\rightarrow e^o_w$
- $p(w) = \frac{\exp(z_w)}{\sum_{w' \in V} \exp(z'_w)}$ where $z_w = h_t^T e^o_w$
- Issue: orthographic confusion
- Solution: Char CNN + whole word embeddings of 128 dimensions ('correction factor')
- Bottleneck layer

Results of replacing output look-up table

Model (RNN state size, e_w^o size)	Test Perplexity	Params (B)
Previous SOTA	51.3	20
LSTM (512, 512)	54.1	0.82
LSTM (1024, 512)	48.2	0.82
LSTM (2048, 512)	43.7	0.83
LSTM (8192, 2048), No dropout	37.9	3.3
LSTM (8192, 2048), Dropout	32.2	3.3
2-layer LSTM (8192, 1024), Big LSTM	30.6	1.8
Big LSTM with CNN Inputs	30.0	1.04
Above with CNN outputs	**39.8**	**0.29**
Above with correction factor	**35.8**	**0.39**

Full character-level language modelling

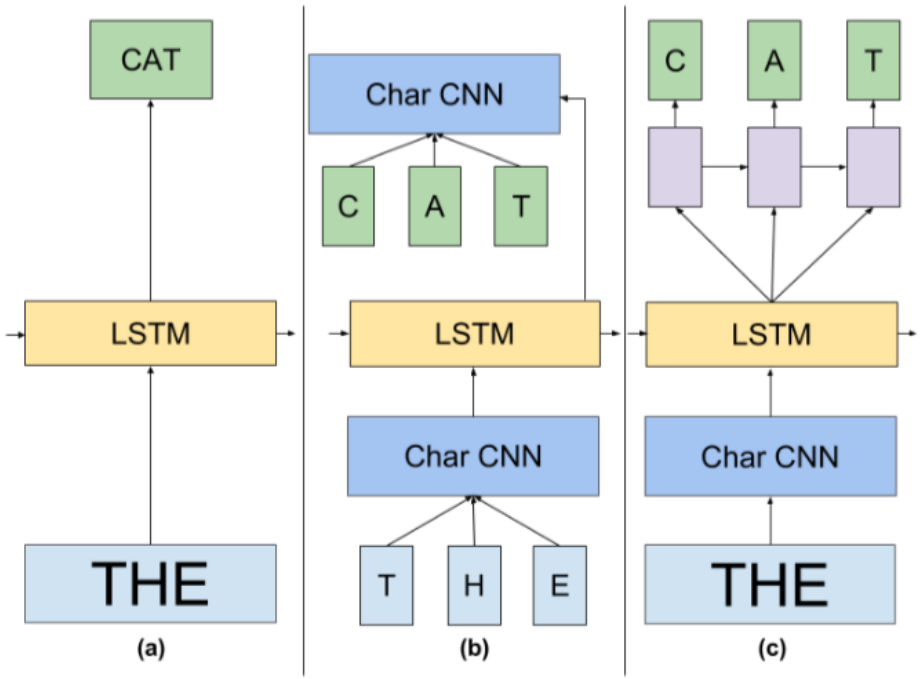


Figure 1. A high-level diagram of the models presented in this paper. (a) is a standard LSTM LM. (b) represents an LM where both input and Softmax embeddings have been replaced by a character CNN. In (c) we replace the Softmax by a next character prediction LSTM network.

Character-level output language modelling

- Replace softmax and output word embeddings with RNN
- RNN conditions on h_t and predicts characters one by one
- Training, word-level model frozen and decoder attached
- Issue: perplexity, $2^{H(P_m)}$
- Solution: Brute force renormalization

Results for character-level output language modelling

Model (RNN state size, e_w^o size)	Test Perplexity	Params (B)
Previous SOTA	51.3	20
LSTM (512, 512)	54.1	0.82
LSTM (1024, 512)	48.2	0.82
LSTM (2048, 512)	43.7	0.83
LSTM (8192, 2048), No dropout	37.9	3.3
LSTM (8192, 2048), Dropout	32.2	3.3
2-layer LSTM (8192, 1024), Big LSTM	30.6	1.8
Big LSTM with CNN Inputs	30.0	1.04
Above with CNN outputs	39.8	0.29
Above with correction factor	35.8	0.39
Big LSTM, characters out	**49.0**	**0.23**
Above with renormalization	**47.9**	**0.23**

Questions?

- **Exploring the Limits of Language Modeling**, Jozefowicz *et al.* 2016
- **Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation**, Ling *et al.* 2015
- **Character-Aware Neural Language Models**, Kim *et al.*, 2015