Predefined Sparseness in Recurrent Sequence Models

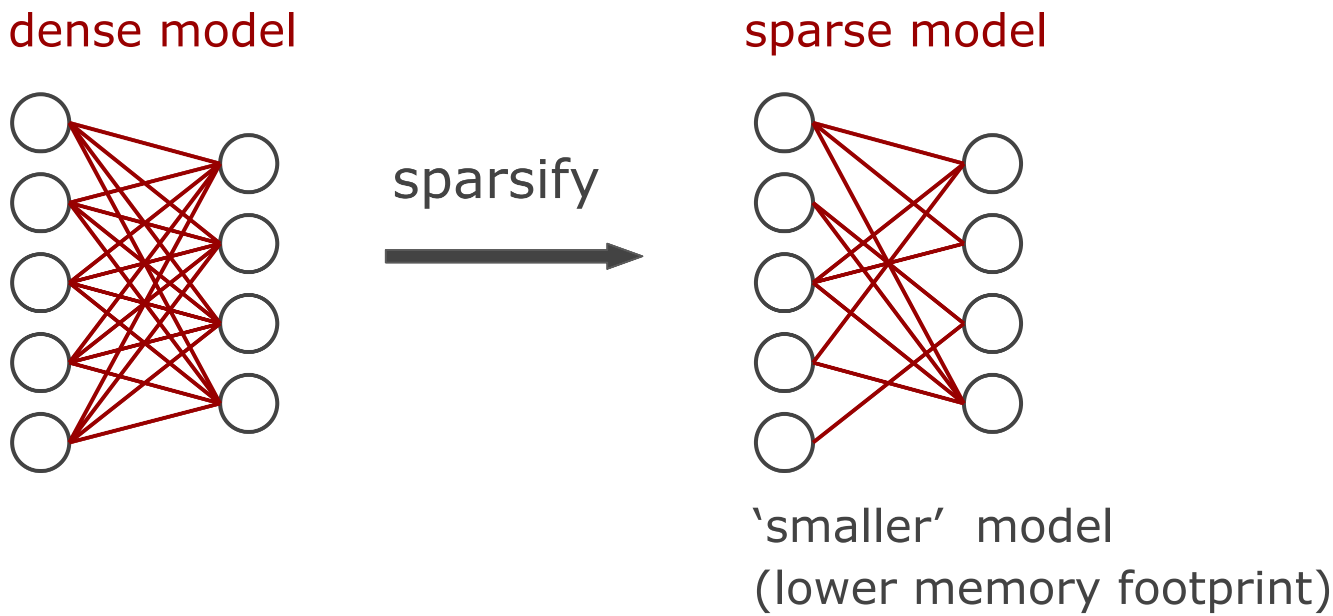
Thomas Demeester, Johannes Deleu, Frederic Godin, Chris Develder

thomas.demeester@ugent.be

GHENT UNIVERSITY

November 1st, 2018
CoNLL, Brussels, Belgium

Sparse Neural Networks



Sparsifying by weight pruning

Highly sparse with accuracy close to dense models [1]

Large sparse networks can be better than small dense models [2]

BUT THEN: large *dense* network needed during training!

GOAL: models that are sparse from the start?

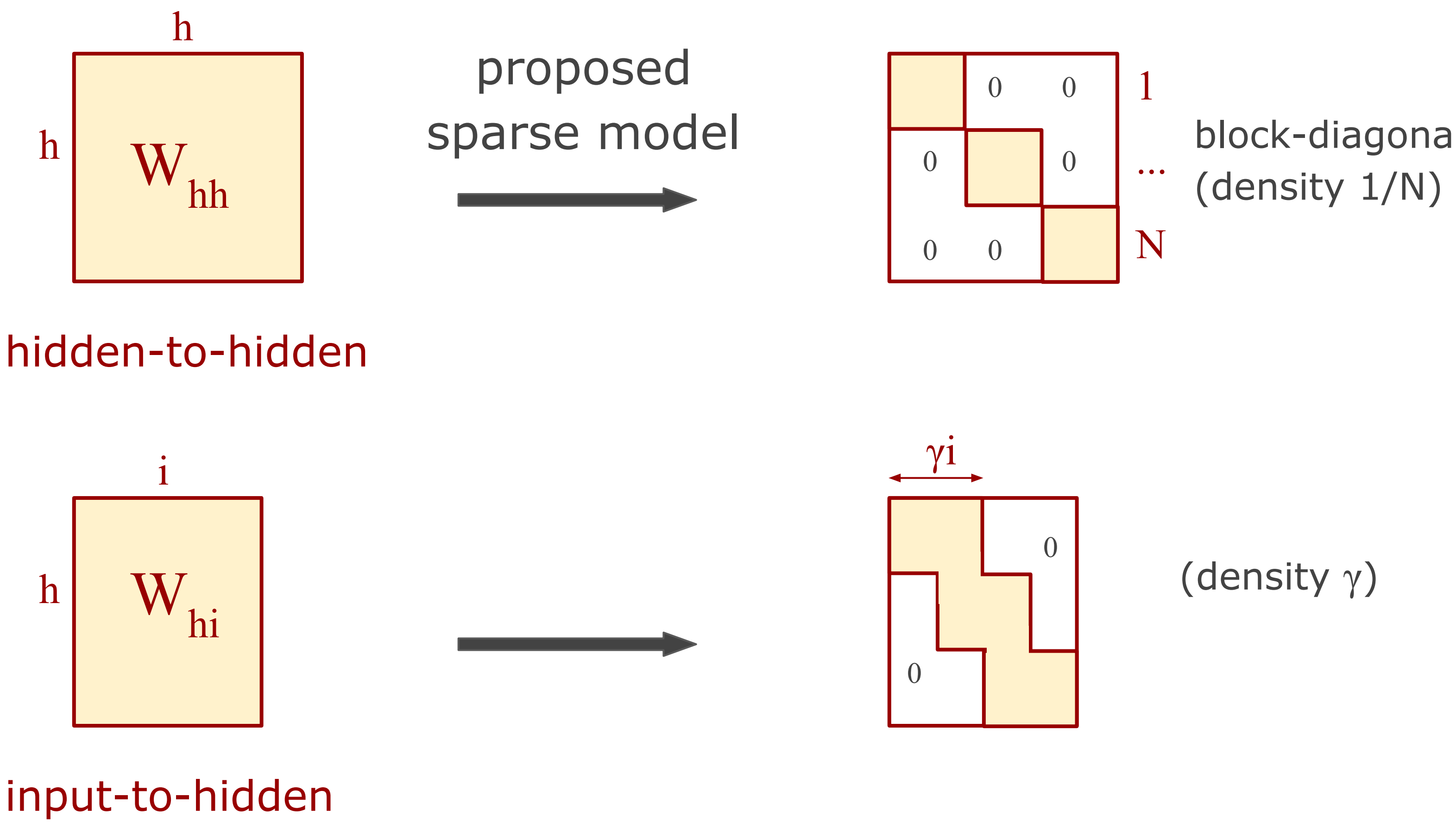
"predefined sparseness"

[1] Narang *et al.* "Exploring Sparsity in RNNs" (ICLR 2017)

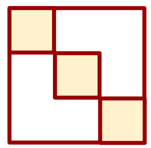
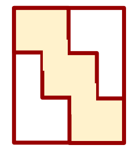
[2] Kalchbrenner *et al.* "Efficient Neural Audio Synthesis" (ICML 2018)

Predefined sparseness for RNNs

Any recurrent cell (RNN, LSTM, GRU...): 2 types of matrices



Predefined sparseness for RNNs

With sparse W_{hh}  and W_{hi} 

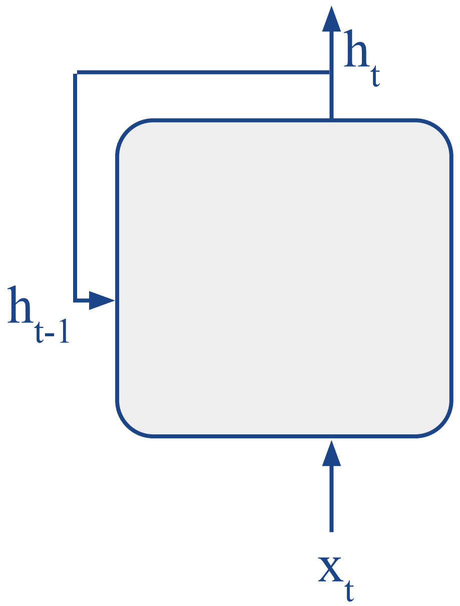
- strongly reduced number of hidden-to-hidden interactions (cfr. weight dropping in W_{hh} [5])
- not all hidden dimensions have access to each input dimension.

why this particular choice?

[5] Merity et al. "Regularizing and Optimizing LSTM Language Models." ICLR 2018.

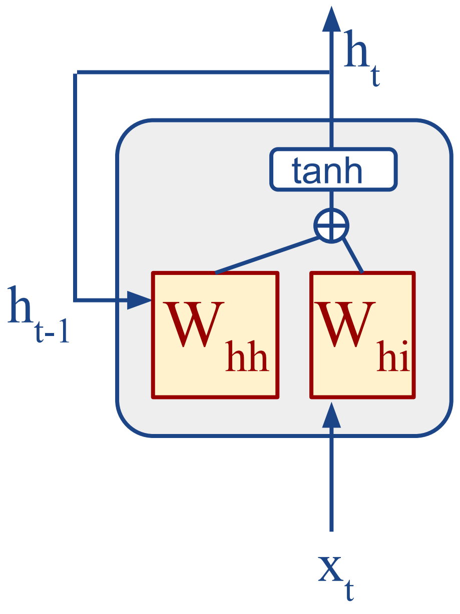
Predefined sparseness for RNNs

Consider vanilla RNN



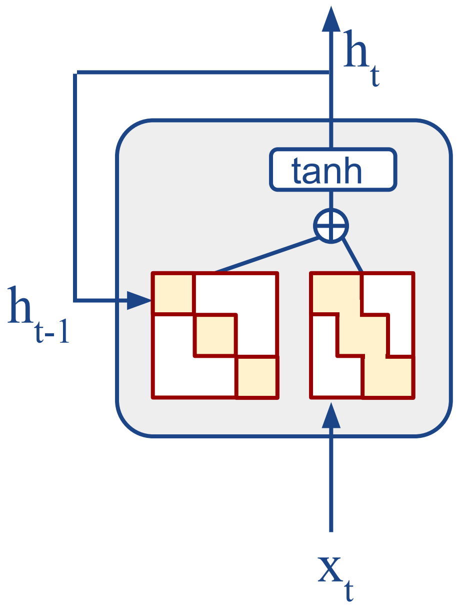
Predefined sparseness for RNNs

Consider vanilla RNN



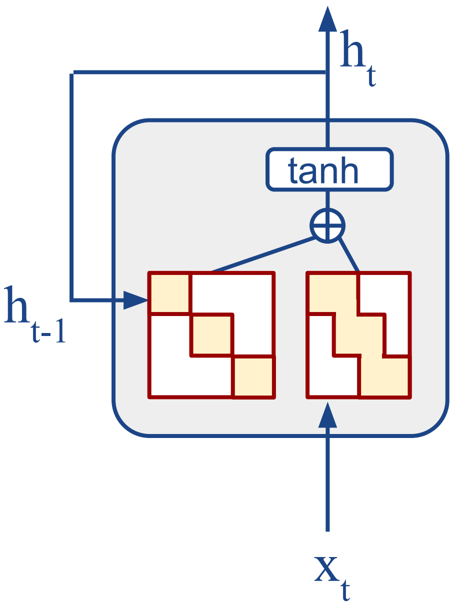
Predefined sparseness for RNNs

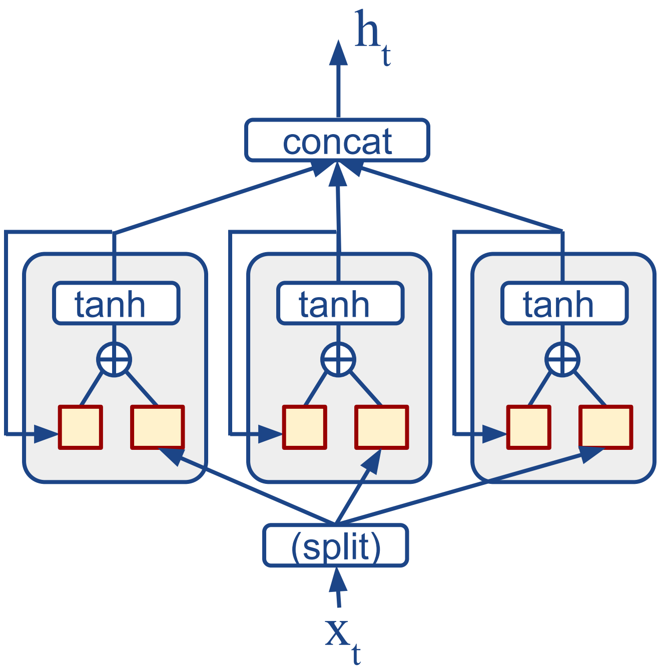
Consider vanilla RNN - made sparse



Predefined sparseness for RNNs

Consider vanilla RNN - made sparse





Resulting RNN equivalent to N smaller dense RNNs in parallel

- only possible with output divided into disjoint segments
- but input can be (partly) shared between components
- holds for vanilla RNN, LSTM, GRU,...
- allows standard tools (CuDNN) / parallel processing

Language modeling with sparse LSTM

- baseline: AWD-LSTM model [5] with 3-layer stacked LSTM

- sparse counterpart:
  - middle LSTM hidden size x 1.5 (from 1150 to 1725)
  - sparse; same number of parameters
  - same regularization settings

[5] Merity et al. "Regularizing and Optimizing LSTM Language Models." ICLR 2018.

Language modeling with sparse LSTM

- first train run (500 epochs)

Model	Penn Treebank test perplexity
reported [5]	58.8
baseline	58.8 ± 0.3
sparse LSTM	57.9 ± 0.3

- train further (‘finetune’) : sparse model overfits

[5] Merity et al. “Regularizing and Optimizing LSTM Language Models.” ICLR 2018.

Language modeling with sparse LSTM

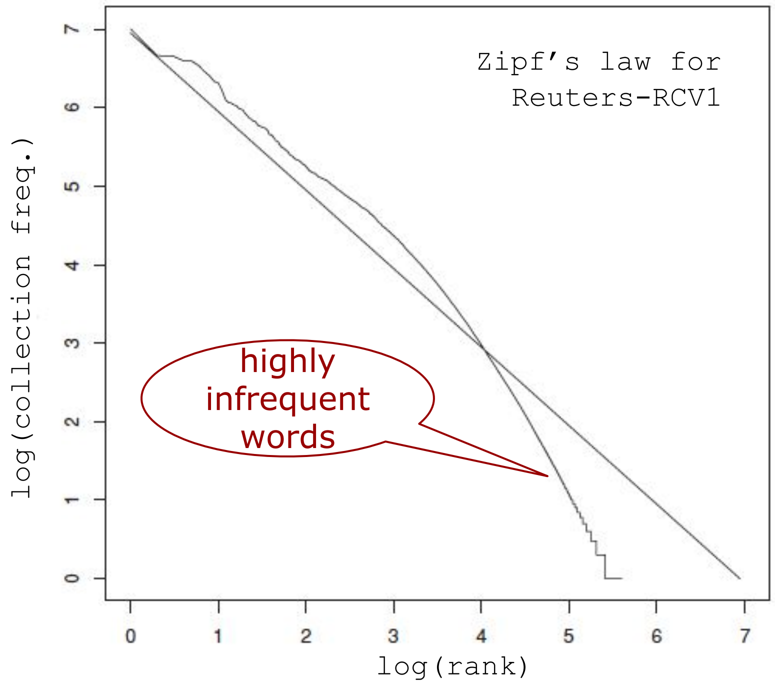
- hypothesis:

  the regularization effect of
  a priori limiting interactions between dimensions
  does not compensate for increased expressiveness
  due to larger hidden state size

- supported by additional experiment “learning to recite” (see paper 😀)

Predefined sparseness in word embeddings

- Goal:
  decide upfront which entries in embedding matrix $\mathbf{E} \in \mathbb{R}^{V\mathrm{x}k}$ are 0.
- Word occurrence frequencies
  have Zipfian nature

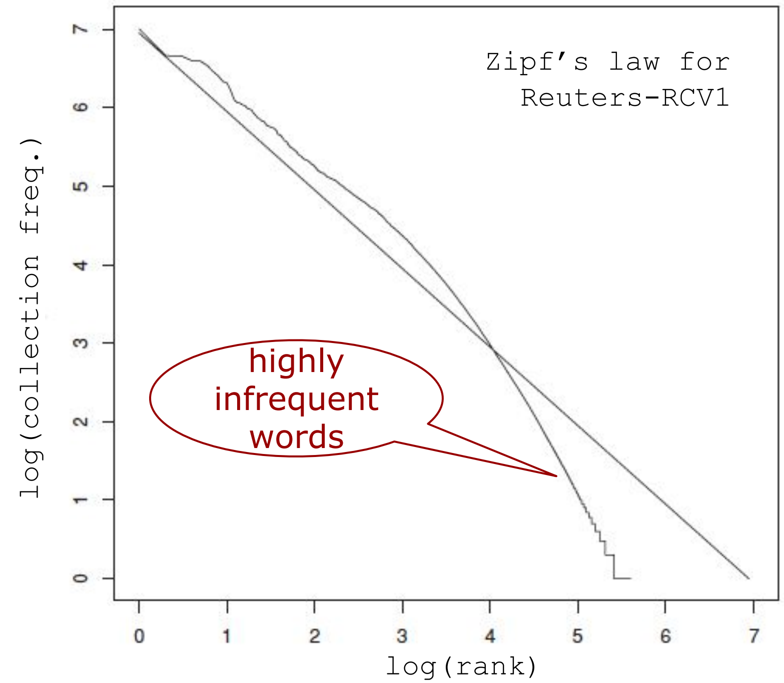


source: Manning, Schütze, Raghavan, "Introduction to Information Retrieval", Cambridge UP, 2009

Predefined sparseness in word embeddings

- Goal:
  decide upfront which entries in embedding matrix $\mathbf{E} \in \mathbb{R}^{V\text{x}k}$ are 0.

- Word occurrence frequencies
  have Zipfian nature

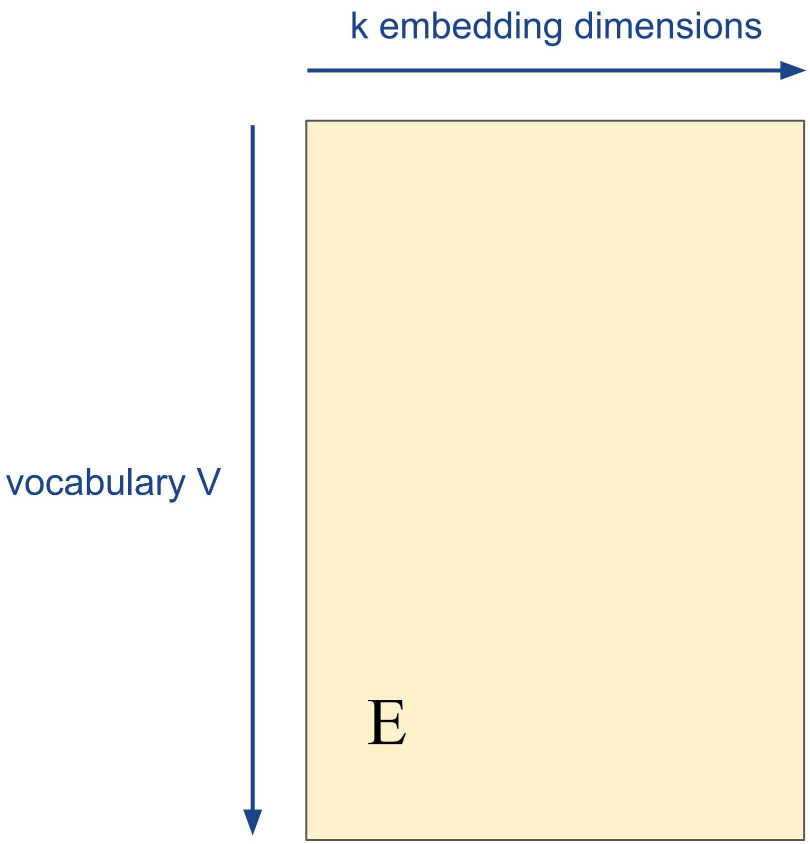


source: Manning, Schütze, Raghavan, "Introduction to Information Retrieval", Cambridge UP, 2009

representing long tail of rare terms with short embeddings would greatly reduce memory requirements

Predefined sparseness in word embeddings

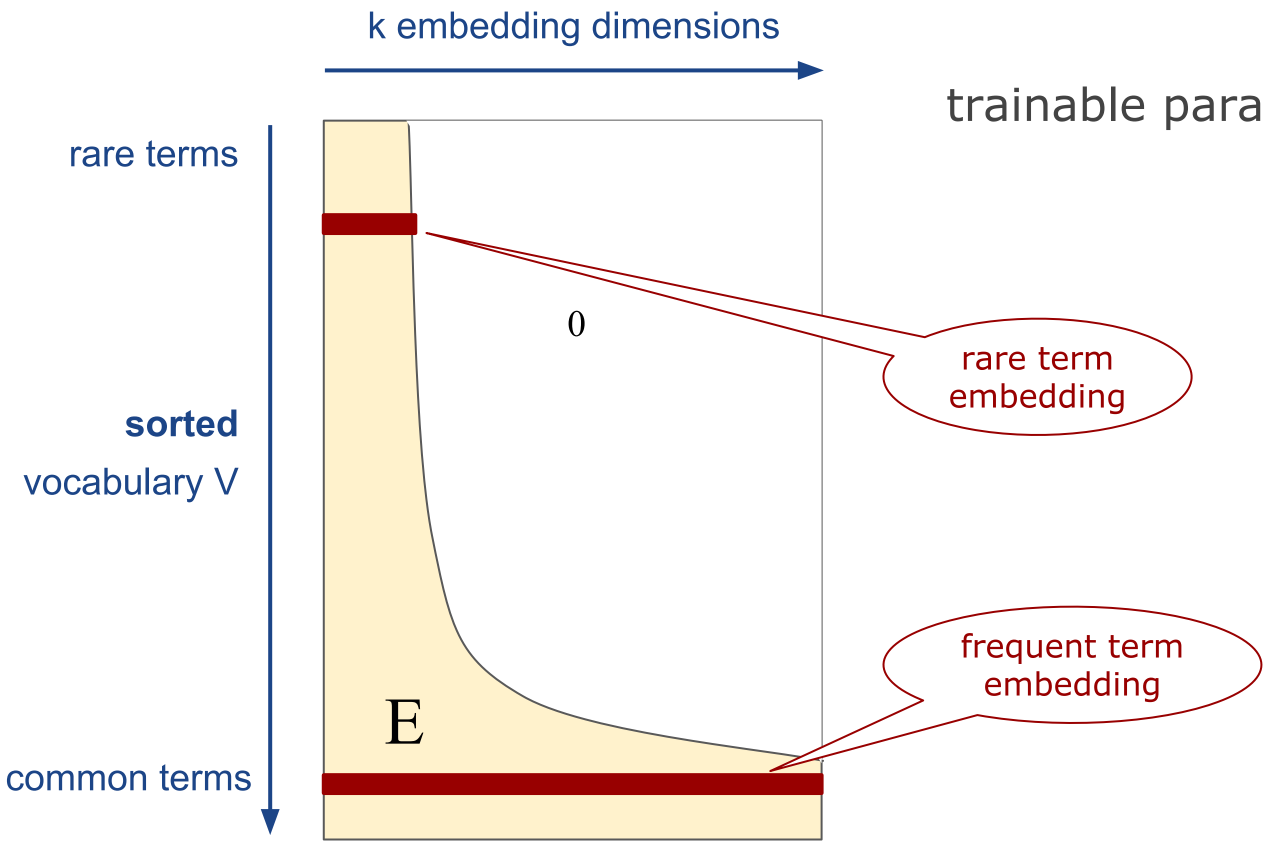
Predefined sparse embedding matrix E ?



trainable parameters $= \mathrm{kV}$

Predefined sparseness in word embeddings

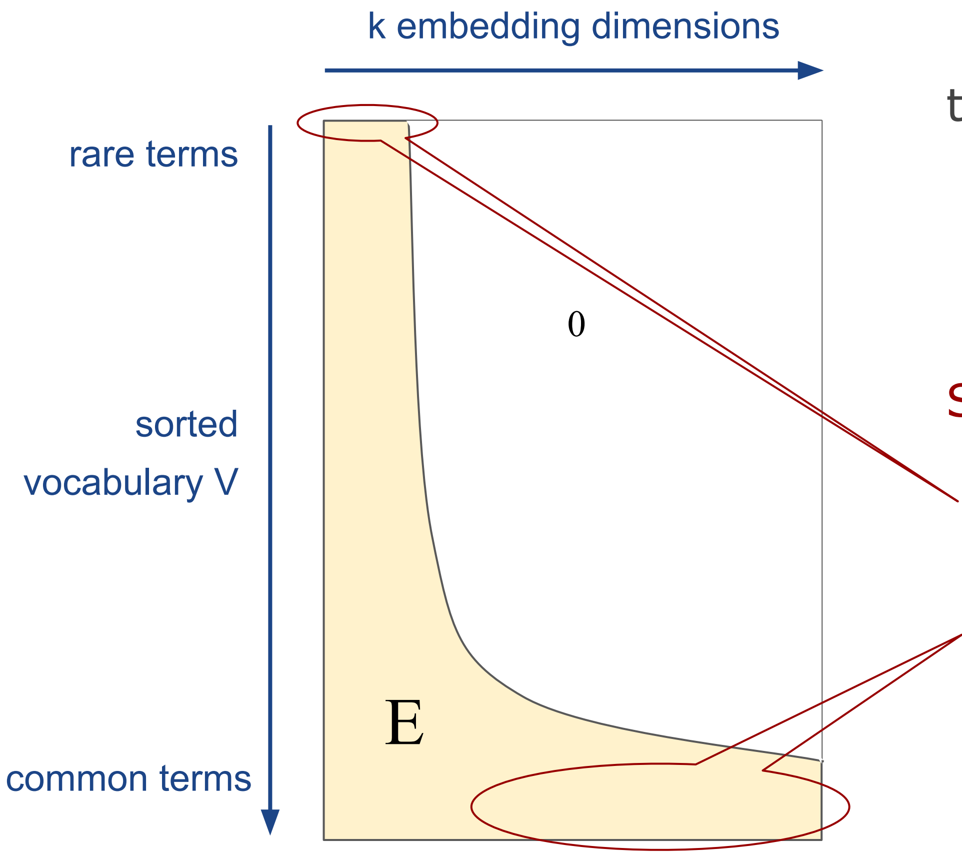
Predefined sparse embedding matrix E?



trainable parameters $= kV\delta_E$

Predefined sparseness in word embeddings

Predefined sparse embedding matrix E?



trainable parameters $= kV\delta_E$

Sparse embedding space

- 'first' dimensions model many rare terms
- remaining dimensions model few frequent terms

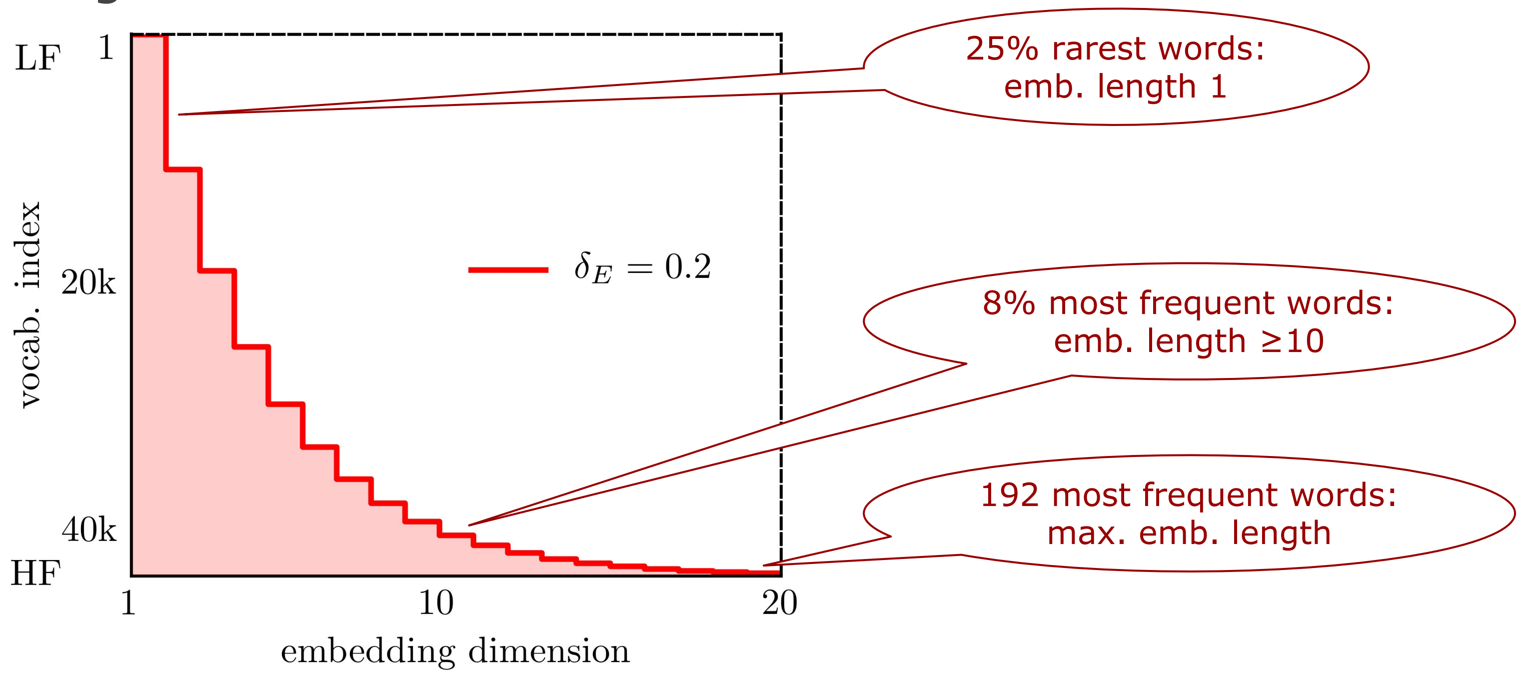
Predefined sparseness in word embeddings

- Experimental setup:
  - POS tagging on Penn Treebank
  - very small model (else too easy!)
  - 20-D word embeddings (876k params)
  - BiLSTM state size 10+10 (3k params)

Predefined sparseness in word embeddings

- Experimental setup:
  - POS tagging on Penn Treebank
  - very small model (else too easy!)
  - 20-D word embeddings (876k params)
  - BiLSTM state size 10+10 (3k params)

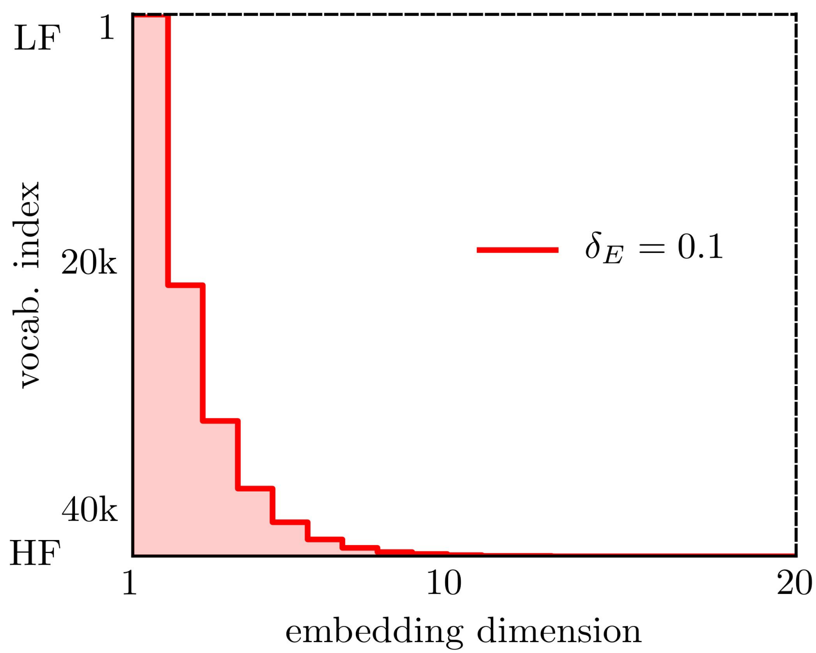
- Embedding matrix



Predefined sparseness in word embeddings

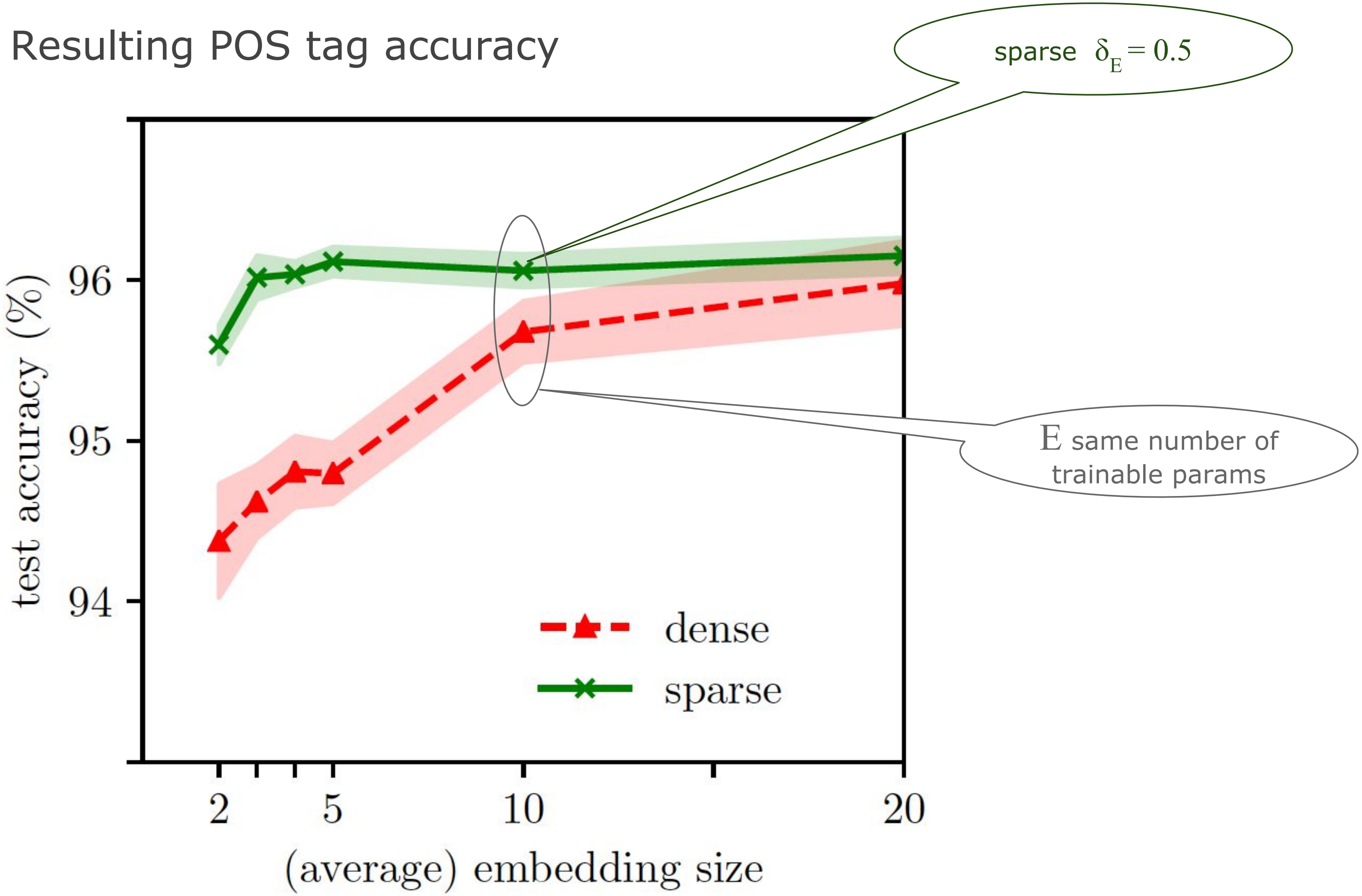
- Experimental setup:
  - POS tagging on Penn Treebank
  - very small model (else too easy!)
  - 20-D word embeddings (876k params)
  - BiLSTM state size 10+10 (3k params)

- Embedding matrix



Predefined sparseness in word embeddings

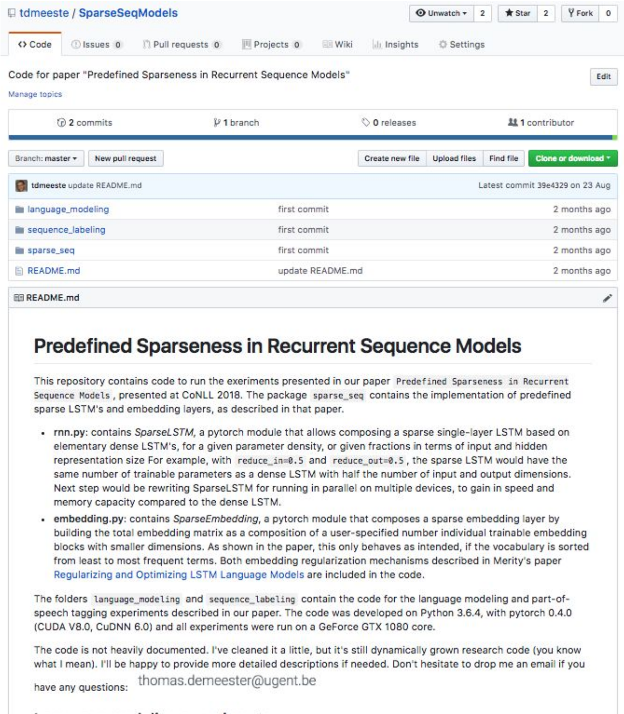
- Resulting POS tag accuracy



Conclusions

- Simple ideas for predefined sparseness in RNNs and embedding layers
- Predefined Sparseness has potential in NLP
- Further investigation needed
  (for very large representation sizes for large vocabularies, etc.)
- Need some "predefined sparseness" code?

  https://github.com/tdmeeste/SparseSeqModels



Thank you!

Language modeling with sparse LSTM

- baseline:
  - AWD-LSTM model [5]
  - 400D word embeddings, 10k words; 4M params
  - 3-layer stacked LSTM (dimensions 400 - 1150 - 400); 20M params

- sparse counterpart:
  - similar 3-layer LSTM; 20M params
  - but: middle LSTM scaled from 1150 to 1725 units (factor 1.5)
    sparse: to retain same number of parameters
  - no tuning (exactly same regularization parameters)

[5] Merity et al. "Regularizing and Optimizing LSTM Language Models." ICLR 2018.

Inspiration from literature

“application of sparse coding in language processing is far from extensive, when compared to speech processing” [3]

Need for sparse models in NLP!

“natural language is high-rank” [4]

How to train large sparse representations despite memory constraints?

[3] Wang et al. “Deep and sparse learning in speech and language processing: An overview.” BICS 2016

[4] Yang et al. “Breaking the softmax bottleneck: a high-rank rnn language model.” ICLR 2018

Language modeling with sparse LSTM

- first train run (500 epochs)

Model	**Penn Treebank test perplexity**
reported [5]	58.8
baseline	58.8 ± 0.3
sparse LSTM	57.9 ± 0.3

- train further (‘finetune’) : sparse model overfits

[5] Merity et al. “Regularizing and Optimizing LSTM Language Models.” ICLR 2018.

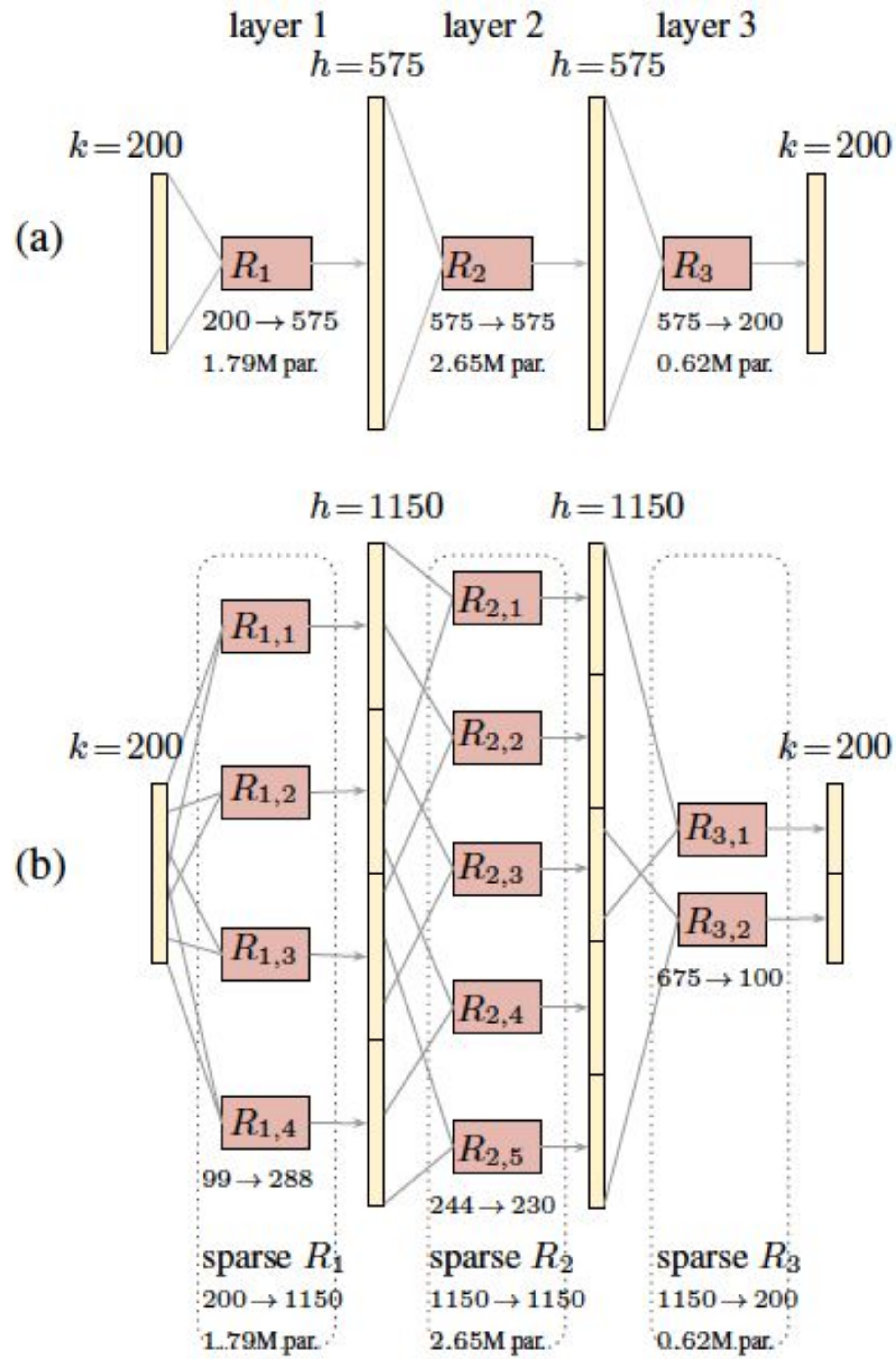
Language modeling with sparse LSTM

- train again (“finetune step” [5])

Model	Penn Treebank test perplexity
reported [5]	57.3
baseline	56.6 ± 0.2
sparse LSTM	57.0 ± 0.2

sparse model
starts overfitting

[5] Merity et al. “Regularizing and Optimizing LSTM Language Models.” ICLR 2018.



layer 1
layer 2
layer 3
h = 575
h = 575
k = 200
k = 200
(a)
R1
R2
R3
200 → 575
1.79M par.
575 → 575
2.65M par.
575 → 200
0.62M par.
h = 1150
h = 1150
k = 200
k = 200
(b)
R1,1
R1,2
R1,3
R1,4
R2,1
R2,2
R2,3
R2,4
R2,5
R3,1
R3,2
99 → 288
244 → 230
675 → 100
sparse R1
200 → 1150
1.79M par.
sparse R2
1150 → 1150
2.65M par.
sparse R3
1150 → 200
0.62M par.