Sequential Language Modeling

Northwestern EECS 474 Probabilistic Graphical Models

Language Modeling

- Modeling sequences of words (today)
  - N-gram Models
  - HMMs (briefly)
  - Neural Network Language Models
- Modeling Documents (next time)
  - "Bag of words"
  - Latent Semantic Analysis, Latent Dirichlet Allocation

Modeling Sequences of Words

- Statistical language models assign **probabilities** to **sequences of words**

P("the dog barked") = $4.203 * 10^{-9}$

- Applications
  - Speech Recognition
  - Machine Translation
  - Spelling Correction
  - Information Extraction

Outline

- N-gram Models
  - Sparsity: Smoothing, Backoff
  - Perplexity Measure
- Exploiting Word Similarity
  - Brown Clustering
  - HMMs (briefly)
  - Neural Network Language Models
- Future Directions

So you want to learn an n-gram model

- Don’t implement one yourself
  - SRILM
    http://www.speech.sri.com/projects/srilm/

Outline

- N-gram Models
  - Sparsity: Smoothing, Backoff
  - **Perplexity Measure**
- Exploiting Word Similarity
  - Brown Clustering
  - HMMs (briefly)
  - Neural Network Language Models
- Future Directions

How to measure LM performance?

- Test model M on held-out text $\bar{w}$ of length *N*

$$\text{Perplexity}(\mathrm{M}, \bar{w}) = 2^{\frac{-\log_2 P_M(\bar{w})}{N}}$$

Models trained on 38 million words from the Wall Street Journal (WSJ) using a 19,979 word vocabulary.

Evaluate on a disjoint set of 1.5 million WSJ words

Perplexity: Trigram < Bigram < Unigram (in this case)

Which N-gram is the best in general?

	Unigram	Bigram	Trigram
Perplexity	962	170	109

http://www.cs.stonybrook.edu/~ychoi/cse628/lecture/02-ngram.pdf

Unigram

Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives

Bigram

Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her

Trigram

They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

Outline

- N-gram Models
  - Sparsity: Smoothing, Backoff
  - Perplexity Measure
- Exploiting Word Similarity
  - **Brown Clustering**
  - HMMs (briefly)
  - Neural Network Language Models
- Future Directions

motivation

Corpus includes:

"Deadline on Monday"

"Deadline on Thursday"

… suggests P("Deadline on ____") ?

Brown Clustering

Say $C : \mathcal{V} \rightarrow \{1, 2, \ldots k\}$ is a *partition* of the vocabulary into k classes

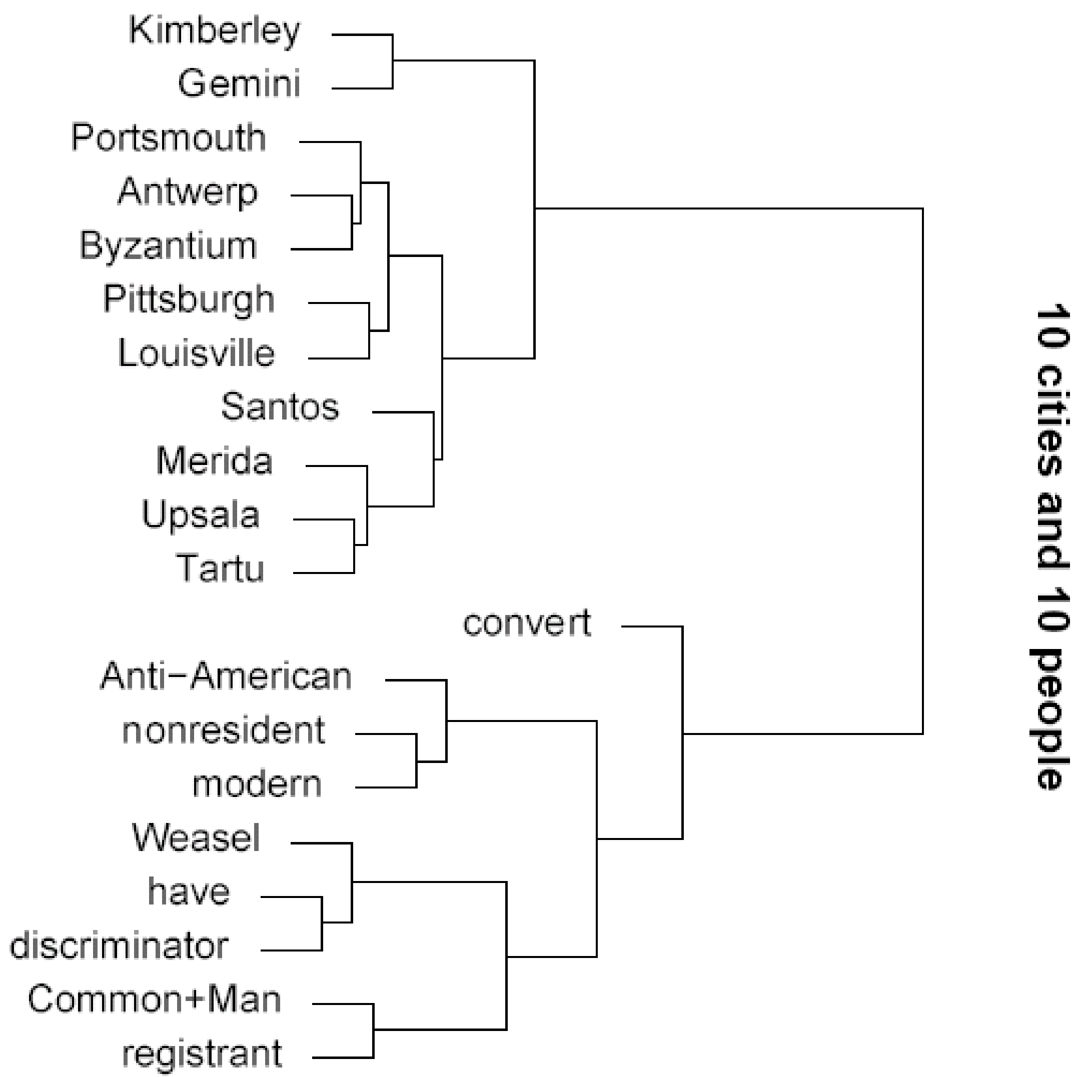
The model:

$$p(w_1, w_2, \ldots w_T) = \prod_{i=1}^{n} p(w_i | C(w_i)) p(C(w_i) | C(w_{i-1}))$$

Michael Collins
http://www.cs.columbia.edu/~cs4705/lectures/brown.pdf

Learning Clusters?

- Greedy Agglomerative Clustering
  - Start with each word in own cluster, iteratively combine the two clusters that results in the smallest decrease in likelihood
  - Slow (cubic in vocab size), faster approaches exist


Kimberley
Gemini
Portsmouth
Antwerp
Byzantium
Pittsburgh
Louisville
Santos
Merida
Upsala
Tartu
convert
Anti−American
nonresident
modern
Weasel
have
discriminator
Common+Man
registrant
10 cities and 10 people

lawyer	1000001101000
newspaperman	100000110100100
stewardess	100000110100101
toxicologist	10000011010011
slang	1000001101010
babysitter	100000110101100
conspirator	1000001101011010
womanizer	1000001101011011
mailman	10000011010111
salesman	100000110110000
bookkeeper	1000001101100010
troubleshooter	10000011011000110
bouncer	10000011011000111
technician	1000001101100100
janitor	1000001101100101
saleswoman	1000001101100110

Miller, Scott, Jethran Guinness, and Alex Zamanian. "Name Tagging with Word Clusters and Discriminative Training." *HLT-NAACL*. Vol. 4. 2004.

Nike	1011011100100101011100
Maytag	10110111001001010111010
Generali	10110111001001010111011
Gap	1011011100100101011110
Harley-Davidson	10110111001001010111110
Enfield	101101110010010101111110
genus	101101110010010101111111
Microsoft	10110111001001011000
Ventritex	101101110010010110010
Tractebel	1011011100100101100110
Synopsys	1011011100100101100111
WordPerfect	1011011100100101101000

Miller, Scott, Jethran Guinness, and Alex Zamanian. "Name Tagging with Word Clusters and Discriminative Training." *HLT-NAACL*. Vol. 4. 2004.

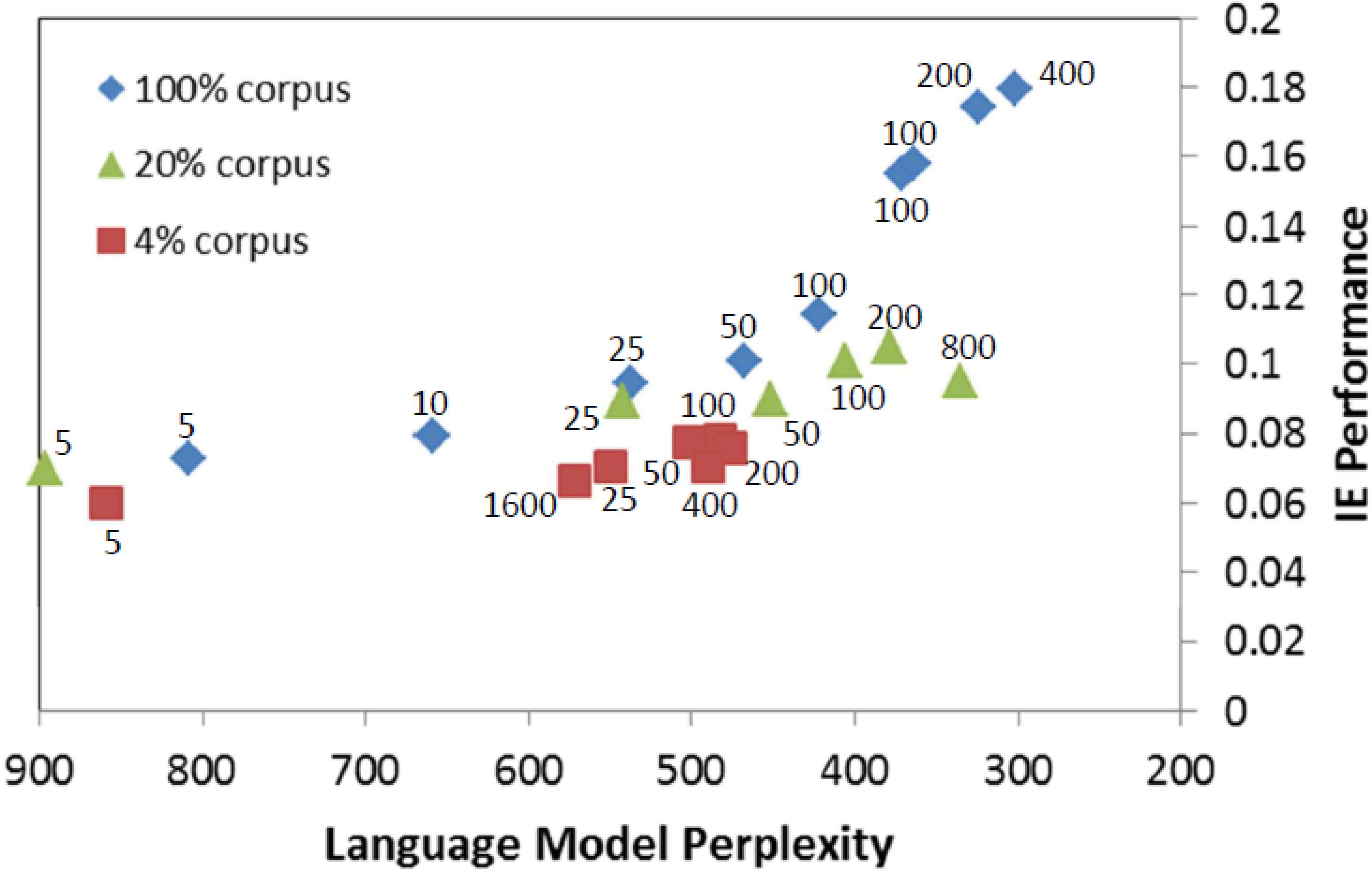
John	101110010000000000
Consuelo	101110010000000001
Jeffrey	101110010000000010
Kenneth	10111001000000001100
Phillip	101110010000000011010
WILLIAM	101110010000000011011
Timothy	10111001000000001110
Terrence	101110010000000011110
Jerald	101110010000000011111
Harold	101110010000000100
Frederic	101110010000000101
Wendell	10111001000000011

Miller, Scott, Jethran Guinness, and Alex Zamanian. "Name Tagging with Word Clusters and Discriminative Training." *HLT-NAACL*. Vol. 4. 2004.

HMMs

- "Soft" version of Brown Clusters
  - as GMMs are to K-means

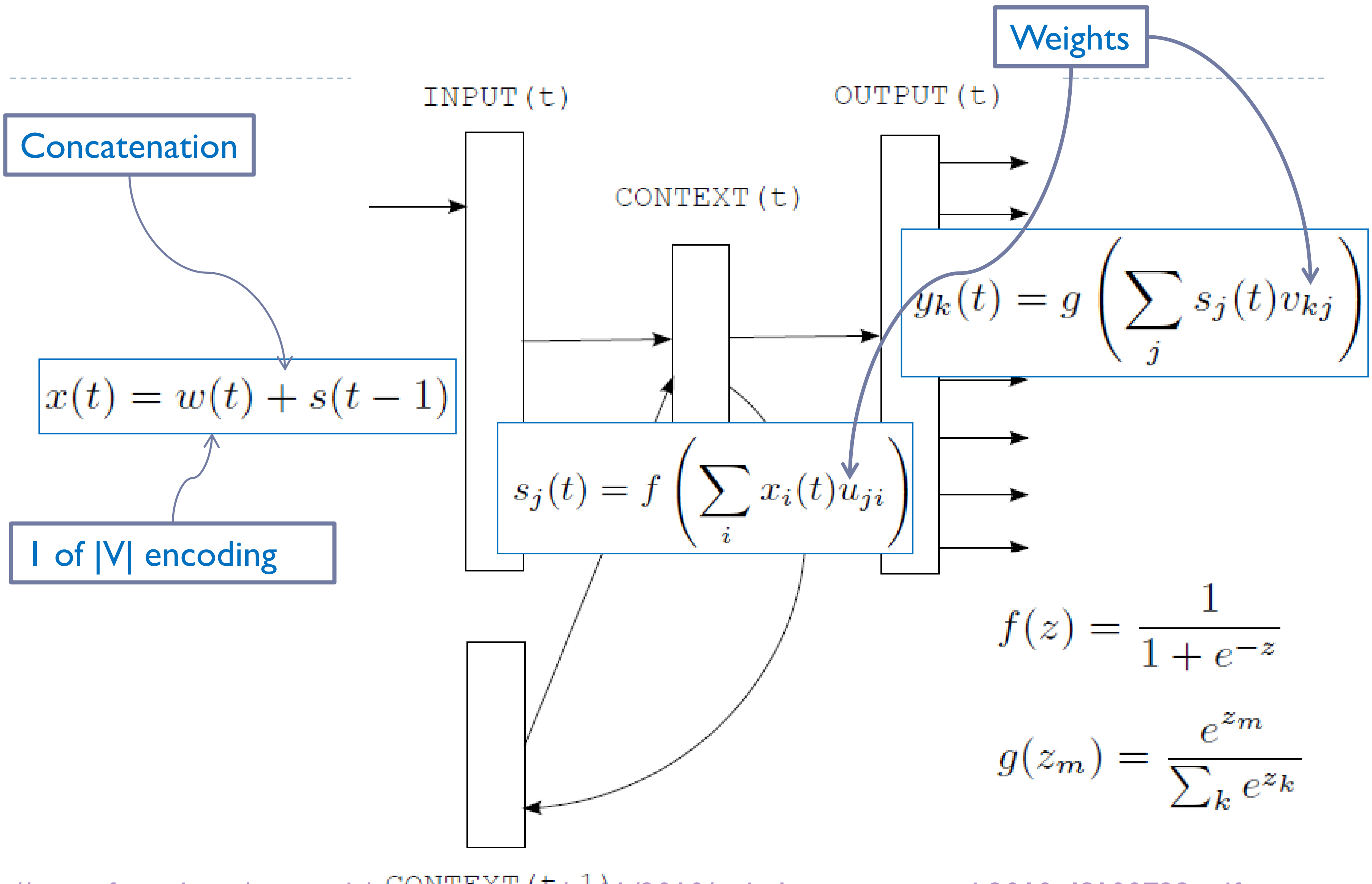
model	AUC
I-HMM-TYPE-R	**0.18**
HMM-TYPE-R	0.17
BROWN-TYPE-R-3200	0.16
BROWN-TYPE-R-1000	**0.18**
BROWN-TYPE-R-320	0.15
BROWN-TYPE-R-100	0.13
LATTICE-TYPE-R	0.11
N-GRAM-R baseline	0.10
Random baseline	0.10



100% corpus
20% corpus
4% corpus
Language Model Perplexity
IE Performance
900
800
700
600
500
400
300
200
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
5
10
25
50
100
100
100
200
400
200
800
1600

Outline

- N-gram Models
  - Sparsity: Smoothing, Backoff
  - Perplexity Measure
- Exploiting Word Similarity
  - Brown Clustering
  - HMMs (briefly)
  - **Neural Network Language Models**
- Future Directions



$$f(z) = \frac{1}{1+e^{-z}}$$

$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$

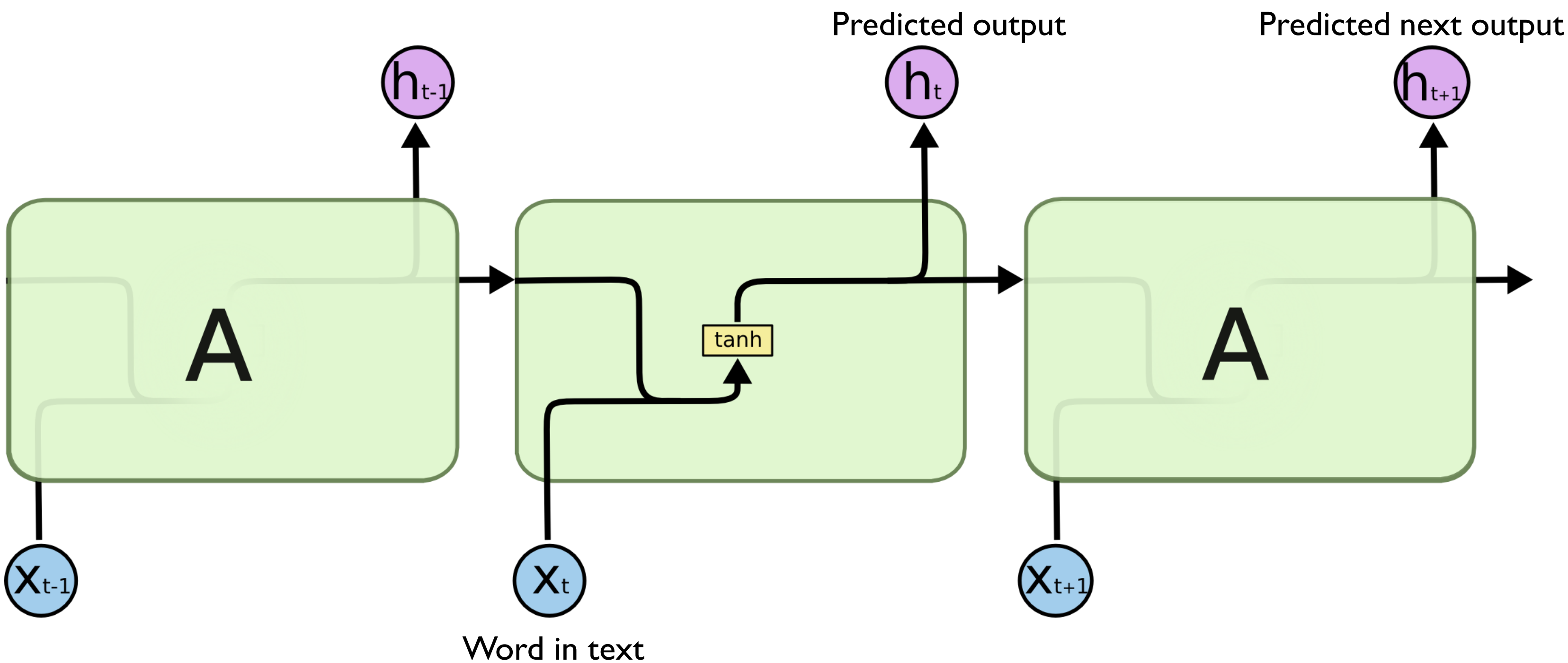
http://www.fit.vutbr.cz/research/groups/speech/publi/2010/mikolov_interspeech2010_IS100722.pdf

Results

0.75*RNN + 0.25 * KN

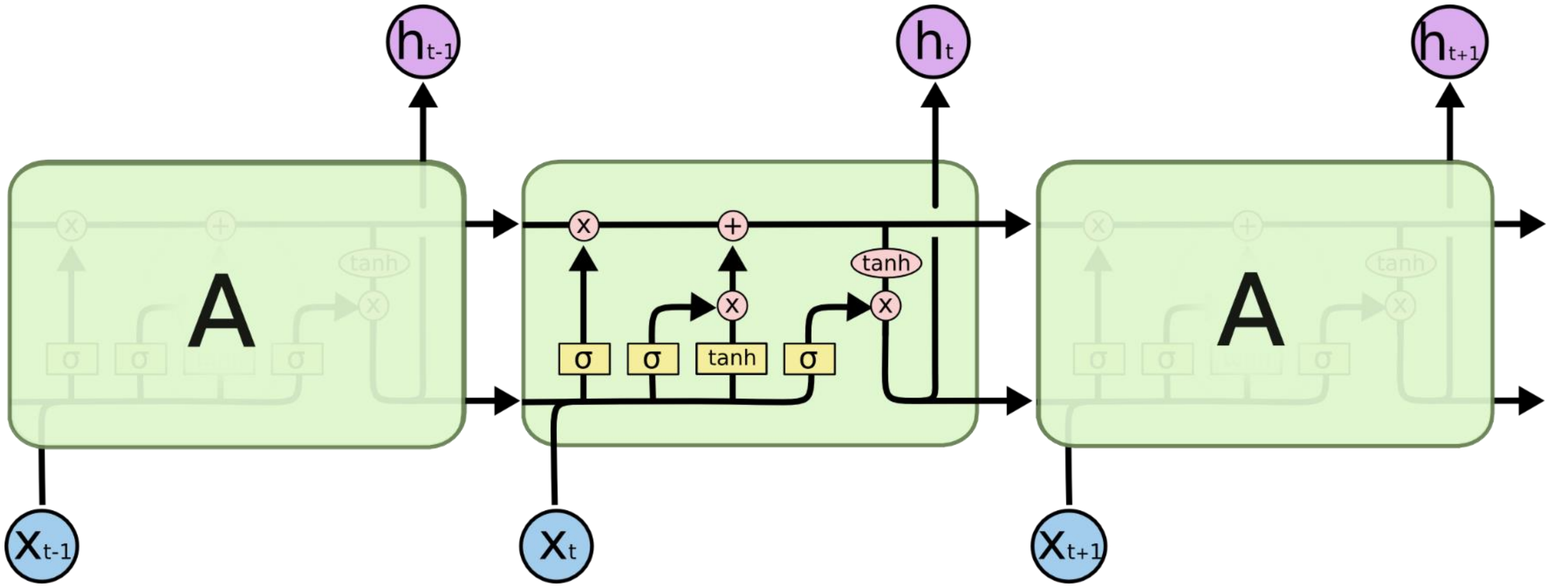
	PPL		WER	
Model	RNN	RNN+KN	RNN	RNN+KN
KN5 - baseline	-	221	-	13.5
RNN 60/20	229	186	13.2	12.6
RNN 90/10	202	173	12.8	12.2
RNN 250/5	173	155	12.3	11.7
RNN 250/2	176	156	12.0	11.9
RNN 400/10	171	152	12.5	12.1
3xRNN static	151	143	11.6	11.3
3xRNN dynamic	128	121	11.3	11.1

RNNs



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTMs



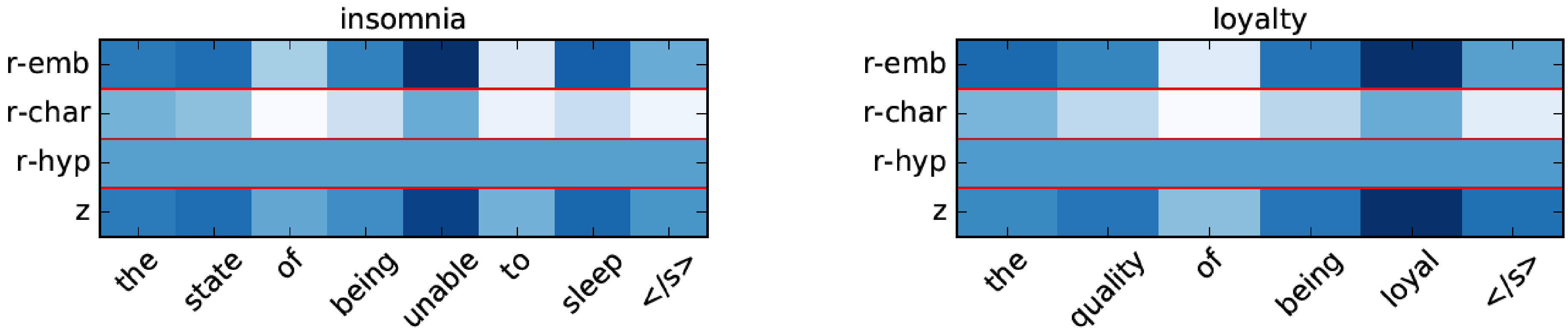
Intuition behind LSTMs

- Harness long-range dependencies
- "She's ___________ so I hope that she will ____________ ."

  brilliant — join us

  sick — not breathe on me

LSTM example



Noraset et al., AAAI 2017

results

- Incredibly good language modeling results
  - Good Turing Smoothing: ~160 perplexity (lower is better)
  - Kneser-Ney smoothing: ~140 perplexity
  - Today's best LSTMs: ~70 perplexity [Google, ca 2015]

Perplexity numbers on Penn TreeBank data set (approx.)