Lecture:
Lexicons and neural networks

CS 585, Fall 2016

Introduction to Natural Language Processing

http://people.cs.umass.edu/~brenocon/inlp2016

Brendan O'Connor

College of Information and Computer Sciences

University of Massachusetts Amherst

This Thursday

https://www.cics.umass.edu/event/improving-memory-capability-recurrent-networks

College of Information and Computer Sciences

ABOUT RESEARCH FUTURE STUDENTS CURRENT STUDENTS PEOPLE

Improving the Memory Capability of Recurrent Networks

Search

29 SEP

Add to Calendar

Thursday, 09/29/2016
2:00pm to 4:00pm

Computer Science Building,
Room 150/151

Statistical and Computational Data Science Distinguished Lecture Series

Speaker: **Yoshua Bengio**

Abstract:

Since the 90s we have known about the fundamental challenge in training a parametrized dynamical system such as a recurrent networks to capture long-term dependencies. The notion of stable memory is crucial in understanding this issue, and is behind the LSTM and GRU architectures, as well as the recent work on networks with an external memory. We present several new ideas exploring how to further expand the reach of recurrent architectures, improve their training and scale up their memory, in particular to model language-related data and better capture semantics for question answering, machine translation and dialogue.

Bio:

- This Thursday: substitute talk for class
  - Exercise: submit 200 word (0.5 pages single spaced) summary to the talk, and one question you have about it.
  - Alternative if you can't make the talk: write about this article by Chris Manning on deep learning and NLP. http://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00239
  - due end of next week
- HW3 posted: due end of next week
- Next week: project discussion

Generalizing linguistic learning

- Build in linguistic knowledge directly
  - Tokenizers define words
  - Keyword lists... don't generalize

- Learn linguistic knowledge from data
  - NB/LogReg: associate words with supervised labels
  - BUT: want to generalize beyond our small labeled dataset!

- Today:
  - *(no ML)* Hand-built sentiment lexicons
  - *(a little ML)* Semi-supervised sentiment lexicons
  - *(lots of ML)* Neural networks

Manually-built sentiment lexicons

- LIWC, MPQA... thousands of words and emotional categories (positive, negative, ...)
- Very commonly used. Created by researchers.
- Better coverage: use lots humans to give judgments

NRC Word-Emotion Association Lexicon

Mohammad and Turney 2011

- 10,000 words chosen mainly from earlier lexicons
- Labeled by Amazon Mechanical Turk
- 5 Turkers per hit
- Give Turkers an idea of the relevant sense of the word
- Result:

```
amazingly   anger   0
amazingly   anticipation    0
amazingly   disgust 0
amazingly   fear    0
amazingly   joy 1
amazingly   sadness 0
amazingly   surprise    1
amazingly   trust   0
amazingly   negative    0
amazingly   positive    1
```

EmoLex	# of terms
EmoLex-Uni:	
Unigrams from Macquarie Thesaurus	
adjectives	200
adverbs	200
nouns	200
verbs	200
EmoLex-Bi:	
Bigrams from Macquarie Thesaurus	
adjectives	200
adverbs	187
nouns	200
verbs	200
EmoLex-GI:	
Terms from General Inquirer	
negative terms	2119
neutral terms	4226
positive terms	1787
EmoLex-WAL:	
Terms from WordNet Affect Lexicon	
anger terms	165
disgust terms	37
fear terms	100
joy terms	165
sadness terms	120
surprise terms	53
Union	**10170**

Mechanical Turk: https://www.mturk.com/mturk/welcome
Crowdsourcing and NLP: http://crowdsourcing-class.org/tutorial.html

The AMT Hit

Prompt word: *startle*

Q1. Which word is closest in meaning (most related) to *startle*?

- *automobile*
- *shake*
- *honesty*
- *entertain*

Q2. How positive (good, praising) is the word *startle*?

- *startle* is not positive
- *startle* is weakly positive
- *startle* is moderately positive
- *startle* is strongly positive

Q3. How negative (bad, criticizing) is the word *startle*?

- *startle* is not negative
- *startle* is weakly negative
- *startle* is moderately negative
- *startle* is strongly negative

Q4. How much is *startle* associated with the emotion joy? (For example, *happy* and *fun* are strongly associated with joy.)

- *startle* is not associated with joy
- *startle* is weakly associated with joy
- *startle* is moderately associated with joy
- *startle* is strongly associated with joy

Q5. How much is *startle* associated with the emotion sadness? (For example, *failure* and *heart-break* are strongly associated with sadness.)

- *startle* is not associated with sadness
- *startle* is weakly associated with sadness
- *startle* is moderately associated with sadness
- *startle* is strongly associated with sadness

Q6. How much is *startle* associated with the emotion fear? (For example, *horror* and *scary* are strongly associated with fear.)

- Similar choices as in 4 and 5 above

Q7. How much is *startle* associated with the emotion anger? (For example, *rage* and *shouting* are strongly associated with anger.)

- Similar choices as in 4 and 5 above

Q8. How much is *startle* associated with the emotion trust? (For example, *faith* and *integrity* are strongly associated with trust.)

- Similar choices as in 4 and 5 above

Q9. How much is *startle* associated with the emotion disgust? (For example, *gross* and *cruelty* are strongly associated with disgust.)

- Similar choices as in 4 and 5 above

...

Semi-supervised learning of lexicons

- Use a small amount of information
  - A few labeled examples
  - A few hand-built patterns
- To bootstrap a lexicon

Turney Algorithm

Turney (2002): Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews

1. Extract a *phrasal lexicon* from reviews
2. Learn polarity of each phrase
3. Rate a review by the average polarity of its phrases

[slides: J&M 3rd ed webpage]

Extract two-word phrases with adjectives

First Word	Second Word	Third Word (not extracted)
JJ	NN or NNS	anything
RB, RBR, RBS	JJ	Not NN nor NNS
JJ	JJ	Not NN or NNS
NN or NNS	JJ	Nor NN nor NNS
RB, RBR, or RBS	VB, VBD, VBN, VBG	anything

[slides: J&M 3rd ed webpage]

How to measure polarity of a phrase?

- Positive phrases co-occur more with *"excellent"*
- Negative phrases co-occur more with *"poor"*
- But how to measure co-occurrence?

[slides: J&M 3rd ed webpage]

Pointwise Mutual Information

- **Mutual information** between 2 random variables X and Y

$$\mathrm{I}(X,Y) = \sum_x \sum_y P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)}$$

- **Pointwise mutual information**:
  - How much more do events x and y co-occur than if they were independent?

$$\mathrm{PMI}(X,Y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

[slides: J&M 3rd ed webpage]

Pointwise Mutual Information

- **Pointwise mutual information**:
  - How much more do events x and y co-occur than if they were independent?

$$\text{PMI}(X,Y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

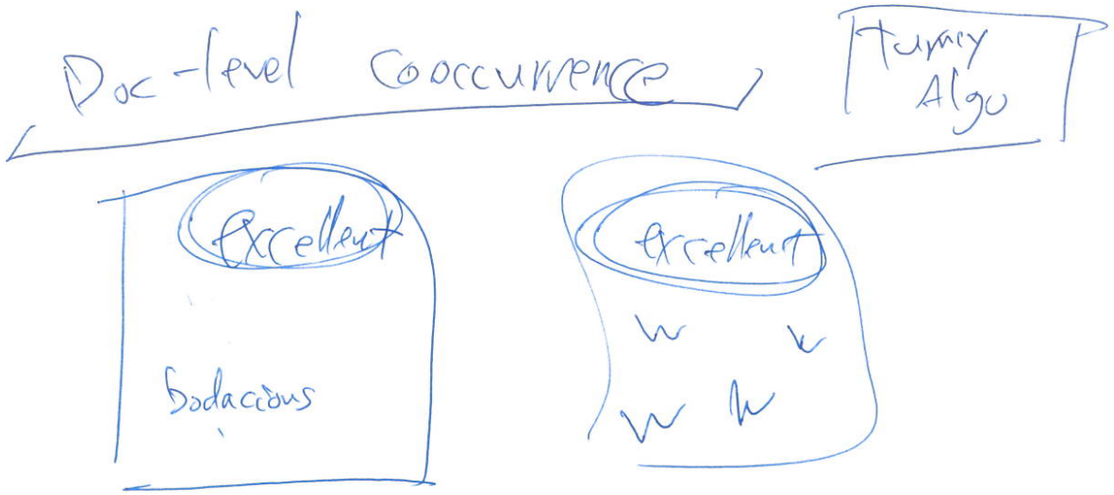
- **PMI between two words**:
  - How much more do two words co-occur than if they were independent?

$$\text{PMI}(word_1, word_2) = \log_2 \frac{P(word_1,word_2)}{P(word_1)P(word_2)}$$

Does phrase appear more with "poor" or "excellent"?

$$\text{Polarity}(phrase) = \text{PMI}(phrase,\text{"excellent"}) - \text{PMI}(phrase,\text{"poor"})$$

Doc-level Cooccurrence



$$PMI(x,y) = \log \frac{p(x,y)}{p(x)\,p(y)}$$

Turney Algo

$$= \log \frac{p(x|y)}{p(x)}$$

$$PMI(\text{lesserevil}, \text{excellent}) = \log \frac{P(\text{le},e)}{P(\text{le})\,P(e)}$$

$$= \text{low}$$

$$PMI(\text{evil}, \text{awesome}) = \log \frac{P(\text{evil}\,|\,\text{awesome})}{P(\text{evil})}$$

Google: [["lesser evil" excellent]] => Count

[["lesser evil"]] => Count

Different Versions of PMI

Contrasting conditional prob vs. background freq

Use counts to calculate PMI (?!)
or just count yourself on corpus

Phrases from a thumbs-up review

Phrase	POS tags	Polarity
online service	JJ NN	2.8
online experience	JJ NN	2.3
direct deposit	JJ NN	1.3
local branch	JJ NN	0.42
…		
low fees	JJ NNS	0.33
true service	JJ NN	-0.73
other bank	JJ NN	-0.85
inconveniently located	JJ NN	-1.5
Average		0.32

[slides: J&M 3rd ed webpage]

Phrases from a thumbs-down review

Phrase	POS tags	Polarity
direct deposits	JJ NNS	5.8
online web	JJ NN	1.9
very handy	RB JJ	1.4
…		
virtual monopoly	JJ NN	-2.0
lesser evil	RBR JJ	-2.3
other problems	JJ NNS	-2.8
low funds	JJ NNS	-6.8
unethical practices	JJ NNS	-8.5
Average		-1.2

[slides: J&M 3rd ed webpage]

Results of Turney algorithm

- 410 reviews from Epinions
  - 170 (41%) negative
  - 240 (59%) positive
- Majority class baseline: 59%
- Turney algorithm: 74%

- Phrases rather than words
- Learns domain-specific information

[slides: J&M 3rd ed webpage]

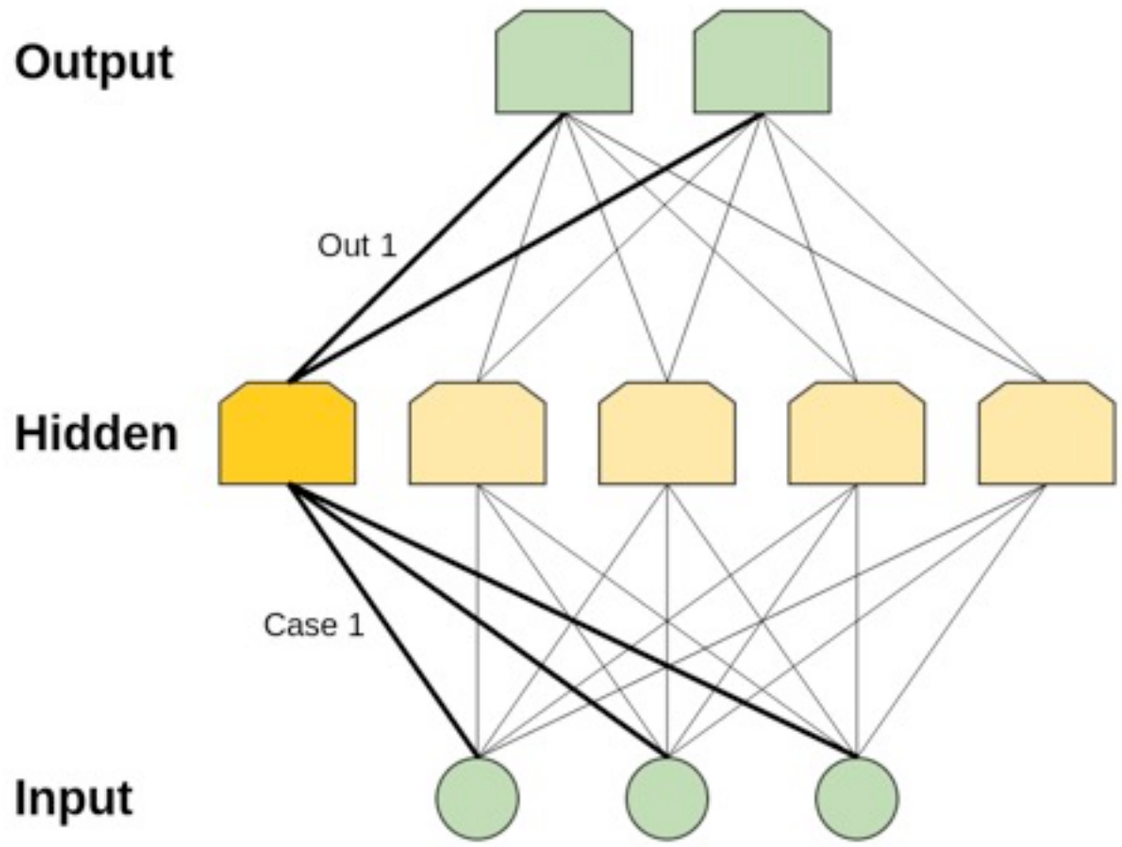
Summary on semi-supervised lexicon learning

- Advantages:
  - Can be domain-specific
  - Can be more robust (more words)
- Intuition
  - Start with a seed set of words (‘good’, ‘poor’)
  - Find other words that have similar polarity:
    - Using “and” and “but”
    - Using words that occur nearby in the same document
    - Using WordNet synonyms and antonyms

[slides: J&M 3rd ed webpage]

Neural networks for NLP

- Neural networks learn latent representations of data/features
  - vs. explicit features
- Feed-forward NN: generalizes logistic regression with **hidden units**



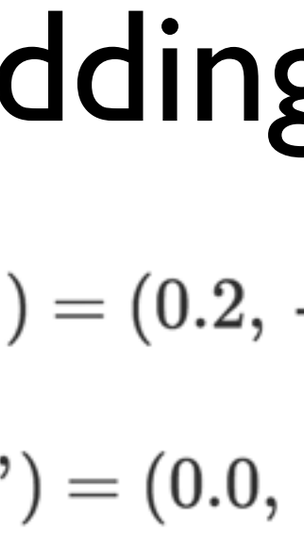
Some nice examples:
http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/

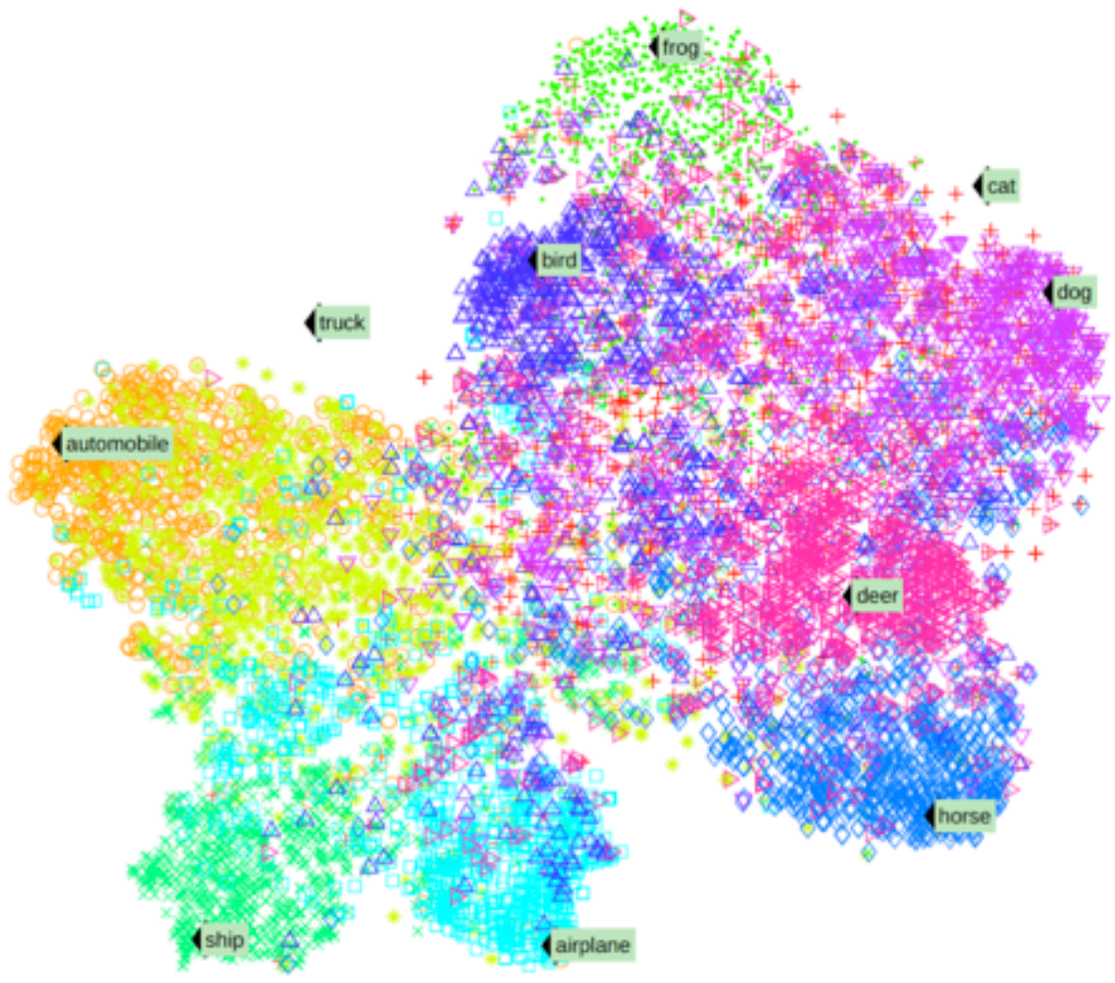
Word embeddings

$$W(\text{“cat”}) = (0.2,\ \text{-}0.4,\ 0.7,\ \ldots)$$

$$W(\text{“mat”}) = (0.0,\ 0.6,\ \text{-}0.1,\ \ldots)$$

- Can be learned as part of a language model or other NLP task
- Often seem to encode word meaning??





Traditional Language Models

- Probability is usually conditioned on window of n previous words
- An incorrect but necessary Markov assumption!

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1}) \approx \prod_{i=1}^{m} P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1})$$

- To estimate probabilities, compute for unigrams and bigrams (conditioning on one/two previous word(s):

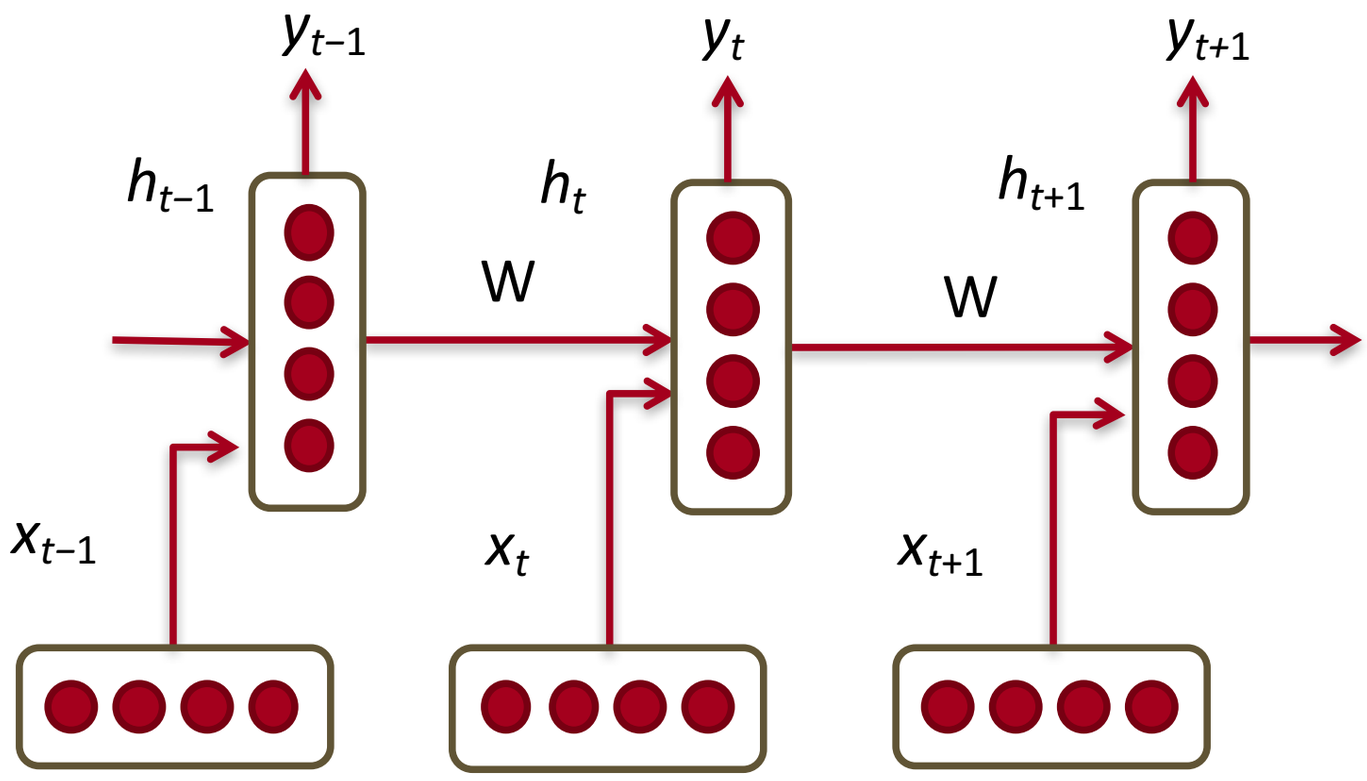
$$p(w_2|w_1) = \frac{\text{count}(w_1, w_2)}{\text{count}(w_1)} \qquad p(w_3|w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

Issues: (1) memory (2) long-distance effects

[slide credit: Richard Socher]

Recurrent Neural Networks!

- RNNs tie the weights at each time step
- Condition the neural network on all previous words
- RAM requirement only scales with number of words



- State of the art LMs
- https://github.com/tensorflow/models/blob/master/lm_1b/README.md

Results

Model	Test Perplexity	Number of Params [billions]
Sigmoid-RNN-2048 [Blackout]	68.3	4.1
Interpolated KN 5-gram, 1.1B n-grams [chelba2013one]	67.6	1.76
Sparse Non-Negative Matrix LM [shazeer2015sparse]	52.9	33
RNN-1024 + MaxEnt 9-gram features [chelba2013one]	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 (50\% Dropout)	32.2	3.3
2-Layer LSTM-8192-1024 (BIG LSTM)	30.6	1.8
(THIS RELEASE) BIG LSTM+CNN Inputs	**30.0**	**1.04**