

# Lecture 10

## Advanced Language Modeling

Michael Picheny, Bhuvana Ramabhadran, Stanley F. Chen,  
Markus Nussbaum-Thom

Watson Group  
IBM T.J. Watson Research Center  
Yorktown Heights, New York, USA  
`{picheny,bhuvana,stanchen,nussbaum}@us.ibm.com`

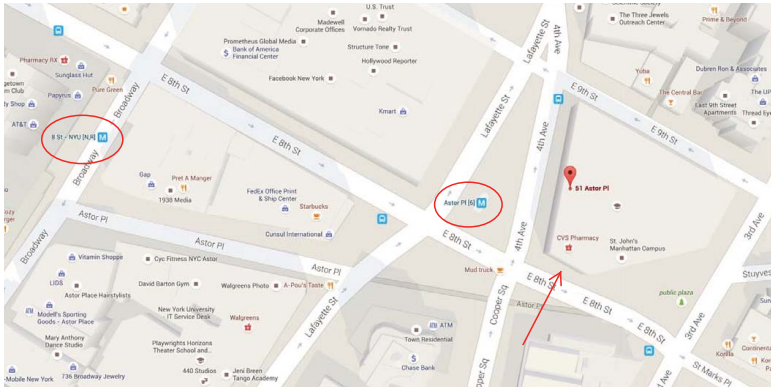
30 March 2016

# Administrivia

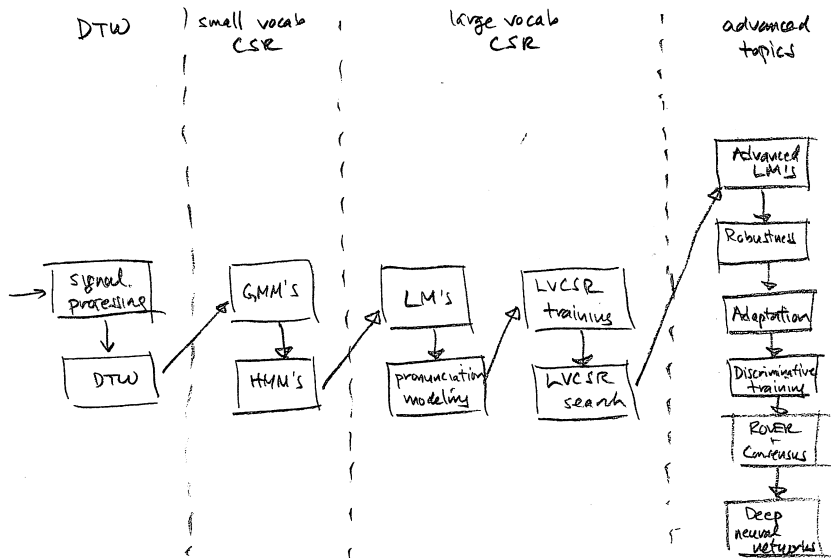
- Lab 3 handed back today?
- Lab 4 extension: due coming Monday, April 4, at 6pm.
- No Lab 5.
- Information on final projects to be announced imminently.

# IBM Watson Astor Place Field Trip!

- In two days: Friday, April 1, 11am-1pm.
- 51 Astor Place; meet in entrance lobby. Free lunch!
- (Going to Watson Client Experience Center on 5th floor.)



# Road Map



# Review: Language Modeling

$$\begin{aligned}(\text{answer}) &= \arg \max_{\omega} (\text{language model}) \times (\text{acoustic model}) \\ &= \arg \max_{\omega} P(\omega)P(\mathbf{x}|\omega)\end{aligned}$$

- Homophones.

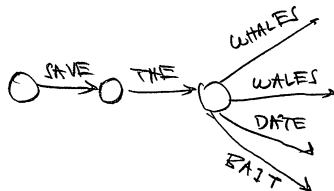
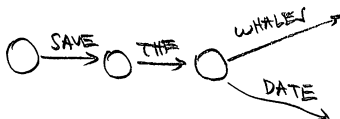
THIS IS OUR ROOM FOR A FOUR HOUR PERIOD .  
THIS IS HOUR ROOM FOUR A FOR OUR . PERIOD

- Confusable sequences in general.

IT IS EASY TO RECOGNIZE SPEECH .  
IT IS EASY TO WRECK A NICE PEACH .

# Language Modeling: Goals

- Assign high probabilities to the good stuff.
- Assign low probabilities to the bad stuff.
  - Restrict choices given to AM.



# Review: $N$ -Gram Models

- Decompose probability of sequence ...
  - Into product of conditional probabilities.
- e.g., trigram model  $\Rightarrow$  Markov order 2  $\Rightarrow$  ...
  - Remember last 2 words.

$$P(\text{I LIKE TO BIKE}) = P(\text{I} | \triangleright \triangleright) \times P(\text{LIKE} | \triangleright \text{I}) \times P(\text{TO} | \text{I LIKE}) \times \\ P(\text{BIKE} | \text{LIKE TO}) \times P(\triangleleft | \text{TO BIKE})$$

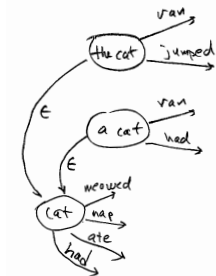
# Estimating N-Gram Models

- Maximum likelihood estimation?

$$P_{\text{MLE}}(\text{TO}|\text{I LIKE}) = \frac{c(\text{I LIKE TO})}{c(\text{I LIKE})}$$

- Smoothing!

$$P_{\text{smooth}}(w_i|w_{i-1}) = \begin{cases} P_{\text{primary}}(w_i|w_{i-1}) & \text{if } \text{count}(w_{i-1} w_i) > 0 \\ \alpha_{w_{i-1}} P_{\text{smooth}}(w_i) & \text{otherwise} \end{cases}$$





# N-Gram Models Are Great!

- $N$ -gram models are robust.
  - Assigns nonzero probs to all word sequences.
- $N$ -gram models are easy to build.
  - Can train on unannotated text; no iteration.
- $N$ -gram models are scalable.
  - Can build 1+ GW models, fast; can increase  $n$ .

# Or Are They?

- In fact,  $n$ -gram models are deeply flawed.
- Let us count the ways.

# What About Short-Distance Dependencies?

BUT THERE'S MORE .PERIOD

IT'S NOT LIMITED TO PROCTER .PERIOD

MR. ANDERS WRITES ON HEALTH CARE FOR THE JOURNAL  
.PERIOD

ALTHOUGH PEOPLE'S PURCHASING POWER HAS FALLEN AND  
SOME HEAVIER INDUSTRIES ARE SUFFERING ,COMMA FOOD  
SALES ARE GROWING .PERIOD

"DOUBLE-QUOTE THE FIGURES BASICALLY SHOW THAT  
MANAGERS HAVE BECOME MORE NEGATIVE TOWARD U. S.  
EQUITIES SINCE THE FIRST QUARTER ,COMMA "DOUBLE-QUOTE  
SAID ANDREW MILLIGAN ,COMMA AN ECONOMIST AT SMITH NEW  
COURT LIMITED .PERIOD

P. & AMPERSAND G. LIFTS PRICES AS OFTEN AS WEEKLY TO  
COMPENSATE FOR THE DECLINE OF THE RUBLE ,COMMA WHICH  
HAS FALLEN IN VALUE FROM THIRTY FIVE RUBLES TO THE  
DOLLAR IN SUMMER NINETEEN NINETY ONE TO THE CURRENT  
RATE OF SEVEN HUNDRED SIXTY SIX .PERIOD

# Poor Generalization

- Lots of unseen  $n$ -grams.
  - *e.g.*, 350MW training  $\Rightarrow$  15% trigrams unseen.
- Seeing “nearby”  $n$ -grams doesn’t help.

LET’S EAT STEAK ON TUESDAY  
LET’S EAT SIRLOIN ON THURSDAY

# Medium-Distance Dependencies?

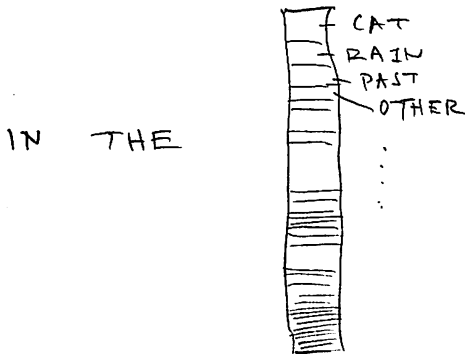
- “Medium-distance”  $\Leftrightarrow$  within utterance.

FABIO WHO WAS NEXT **ASKED** IF THE TELLER...

- Does a trigram model do the “right” thing?

# Generating Text From A Language Model

- Reveals what word sequences model thinks is likely.
- e.g.,  $P(w_i | \text{IN THE})$



# Trigram Model Trained On WSJ, 20MW

AND WITH WHOM IT MATTERS AND IN THE SHORT -HYPHEN TERM  
AT THE UNIVERSITY OF MICHIGAN IN A GENERALLY QUIET SESSION  
THE STUDIO EXECUTIVES LAW  
REVIEW WILL FOCUS ON INTERNATIONAL UNION OF THE STOCK MARKET  
HOW FEDERAL LEGISLATION  
"DOUBLE-QUOTE SPENDING  
THE LOS ANGELES  
THE TRADE PUBLICATION  
SOME FORTY %PERCENT OF CASES ALLEGING GREEN PREPARING FORMS  
NORTH AMERICAN FREE TRADE AGREEMENT (LEFT-PAREN NAFTA  
)RIGHT-PAREN ,COMMA WOULD MAKE STOCKS  
A MORGAN STANLEY CAPITAL INTERNATIONAL PERSPECTIVE ,COMMA GENEVA  
"DOUBLE-QUOTE THEY WILL STANDARD ENFORCEMENT  
THE NEW YORK MISSILE FILINGS OF BUYERS

- What's wrong?

# What Are Real Utterances Like?

- Don't end/start abruptly.
- Have matching quotes.
- Are about single subject.
- May even be grammatical.
- And make sense.
- Why can't  $n$ -gram models model this stuff?



# Long-Distance Dependencies?

- “Long-distance”  $\Leftrightarrow$  between utterance.
- $P(\omega = w_1 \cdots w_I)$  = frequency of utterance?
- $P(\vec{\omega} = \omega_1 \cdots \omega_L)$  = frequency of utterance sequence!

# Trigram Model Trained On WSJ, 20MW

AND WITH WHOM IT MATTERS AND IN THE SHORT -HYPHEN TERM  
AT THE UNIVERSITY OF MICHIGAN IN A GENERALLY QUIET SESSION  
THE STUDIO EXECUTIVES LAW  
REVIEW WILL FOCUS ON INTERNATIONAL UNION OF THE STOCK MARKET  
HOW FEDERAL LEGISLATION  
"DOUBLE-QUOTE SPENDING  
THE LOS ANGELES  
THE TRADE PUBLICATION  
SOME FORTY %PERCENT OF CASES ALLEGING GREEN PREPARING FORMS  
NORTH AMERICAN FREE TRADE AGREEMENT (LEFT-PAREN NAFTA  
)RIGHT-PAREN ,COMMA WOULD MAKE STOCKS  
A MORGAN STANLEY CAPITAL INTERNATIONAL PERSPECTIVE ,COMMA GENEVA  
"DOUBLE-QUOTE THEY WILL STANDARD ENFORCEMENT  
THE NEW YORK MISSILE FILINGS OF BUYERS

- What's wrong?

# What Is Real Text Like?

- Adjacent utterances tend to be on same topic.
- And refer to same entities, *e.g.*, Clinton.
- In a similar style, *e.g.*, formal *vs.* conversational.
- Why can't  $n$ -gram models model this stuff?

# Recap: Shortcomings of $N$ -Gram Models

- Not great at modeling short-distance dependencies.
- Not great at modeling medium-distance dependencies.
- Not great at modeling long-distance dependencies.
- Basically, dumb idea.
  - Insult to language modeling researchers.
  - Great for me to poop on.
  - $N$ -gram models, . . . you're fired!

# Part I

## Language Modeling, Pre-2005-ish

# Where Are We?

- 1 Short-Distance Dependencies: Word Classes
- 2 Medium-Distance Dependencies: Grammars
- 3 Long-Distance Dependencies: Adaptation
- 4 Decoding With Advanced Language Models
- 5 Discussion

# Improving Short-Distance Modeling

- Word  $n$ -gram models do not generalize well.

LET'S EAT STEAK ON TUESDAY  
LET'S EAT SIRLOIN ON THURSDAY

- Idea: word  $n$ -gram  $\Rightarrow$  class  $n$ -grams!?

$$P_{\text{MLE}}([\text{DAY}] \mid [\text{FOOD}] [\text{PREP}]) = \frac{c([\text{FOOD}] [\text{PREP}] [\text{DAY}])}{c([\text{FOOD}] [\text{PREP}])}$$

- Any instance of class trigram increases ...
  - Probs of all other instances of class trigram.

# Getting From Class to Word Probabilities

- What we have:

$$P([\text{DAY}] \mid [\text{FOOD}] [\text{PREP}]) \Leftrightarrow P(c_i \mid c_{i-2} c_{i-1})$$

- What we want:

$$P(\text{THURSDAY} \mid \text{SIRLOIN ON}) \Leftrightarrow P(w_i \mid w_{i-2} w_{i-1})$$

- Predict current word given (non-hidden) class.

$$\begin{aligned} P(\text{THURSDAY} \mid \text{SIRLOIN ON}) = \\ P([\text{DAY}] \mid [\text{FOOD}] [\text{PREP}]) \times P(\text{THURSDAY} \mid [\text{DAY}]) \end{aligned}$$



# How To Assign Words To Classes?

- For generalization to work sensibly ...
  - Group “related” words in same class.

ROSE FELL DROPPED GAINED JUMPED CLIMBED SLIPPED  
HEYDAY MINE'S STILL MACHINE NEWEST HORRIFIC BEECH

- With vocab sizes of 50,000+, can't do this manually.
  - $\Rightarrow$  Unsupervised clustering!

# Word Clustering (Brown *et al.*, 1992)

- Class trigram model.

$$P(w_i | w_{i-2} w_{i-1}) = P(c_i | c_{i-2} c_{i-1}) \times P(w_i | c_i)$$

- Idea: choose classes to optimize training likelihood (MLE)!?
- Simplification: use class **bigram** model.

$$P(w_i | w_{i-1}) = P(c_i | c_{i-1}) \times P(w_i | c_i)$$

- Fix number of classes, *e.g.*, 1000; hill-climbing search.

# Example Classes, 900MW Training Data

OF

THE TONIGHT'S SARAJEVO'S JUPITER'S PLATO'S CHILDHOOD'S  
GRAVITY'S EVOLUTION'S

AS BODES AUGURS BODED AUGURED

HAVE HAVEN'T WHO'VE

DOLLARS BARRELS BUSHEL DOLLARS' KILOLITERS

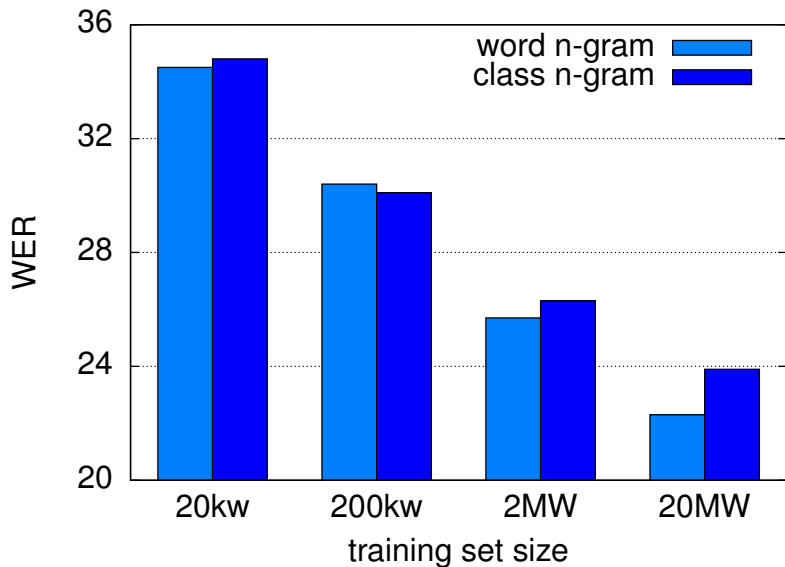
MR. MS. MRS. MESSRS. MRS

HIS SADDAM'S MOZART'S CHRIST'S LENIN'S NAPOLEON'S JESUS'  
ARISTOTLE'S DUMMY'S APARTHEID'S FEMINISM'S

ROSE FELL DROPPED GAINED JUMPED CLIMBED SLIPPED TOTALED  
EASED PLUNGED SOARED SURGED TOTALING AVERAGED TUMBLED

SLID SANK SLUMPED REBOUNDED PLUMMETED DIPPED FIRMED  
RETREATED TOTALLING LEAPED SHRANK SKIDDED ROCKETED SAGGED  
LEAPT ZOOMED SPURTED RALLIED TOTALLED NOSEDIVED

# Class $N$ -Gram Model Performance (WSJ)



# Combining Models: Linear Interpolation

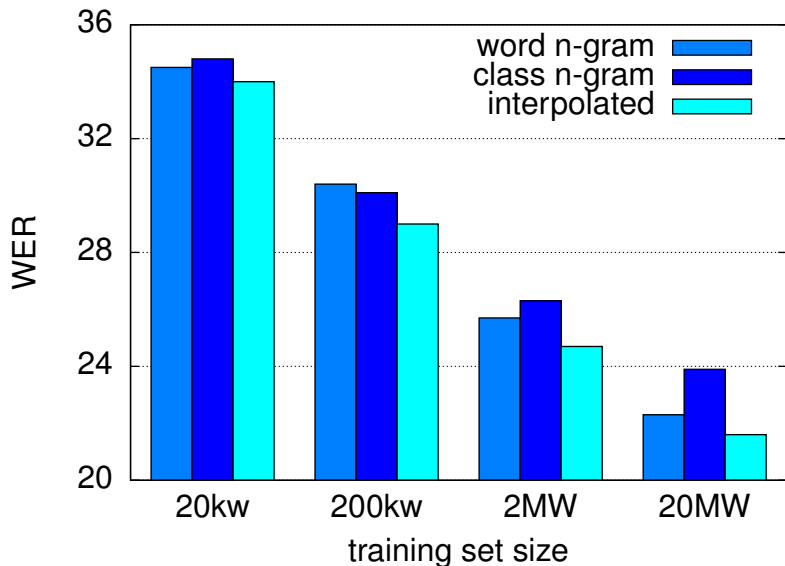
- A “hammer” for combining models.

$$P_{\text{combine}}(\cdot|\cdot) = \lambda \times P_1(\cdot|\cdot) + (1 - \lambda) \times P_2(\cdot|\cdot)$$

- Combined model probabilities sum to 1 correctly.
- Easy to train  $\lambda$  to maximize likelihood of data. (How?)

$$P_{\text{combine}}(w_i | w_{i-2} w_{i-1}) = \lambda \times P_{\text{word}}(w_i | w_{i-2} w_{i-1}) + \\ (1 - \lambda) \times P_{\text{class}}(w_i | w_{i-2} w_{i-1})$$

# Combining Word and Class N-Gram Models



# Discussion: Class $N$ -Gram Models

- Smaller than word  $n$ -gram models.
  - $N$ -gram model over vocab of  $\sim 1000$ , not  $\sim 50000$ .
  - Interpolation  $\Rightarrow$  overall model larger.
- Easy to add new words to vocabulary.
  - Only need to initialize  $P(w_{\text{new}} \mid c_{\text{new}})$ .

$$P(w_i \mid w_{i-2} w_{i-1}) = P(c_i \mid c_{i-2} c_{i-1}) \times P(w_i \mid c_i)$$

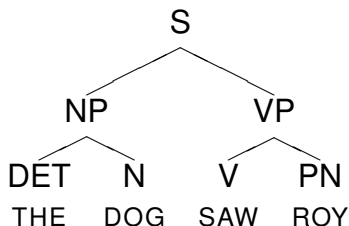
# Where Are We?

- 1 Short-Distance Dependencies: Word Classes
- 2 Medium-Distance Dependencies: Grammars
- 3 Long-Distance Dependencies: Adaptation
- 4 Decoding With Advanced Language Models
- 5 Discussion



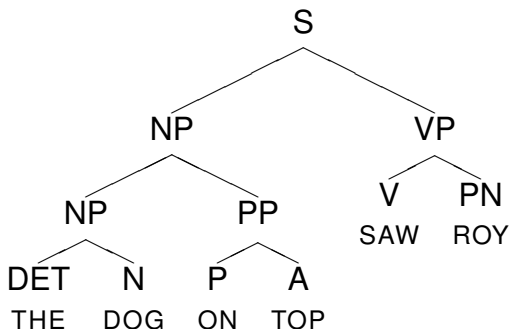
# Modeling Medium-Distance Dependencies

- *N*-gram models predict identity of next word ...
  - Based on identities of words in fixed positions in past.
- Important words for prediction may occur elsewhere.
  - Important word for predicting SAW is DOG.



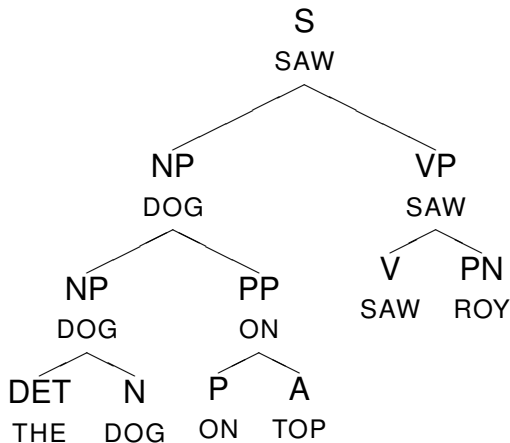
# Modeling Medium-Distance Dependencies

- Important words for prediction may occur elsewhere.
  - Important word for predicting SAW is DOG.
- Instead of condition on fixed number of words back ...
  - Condition on words in fixed positions in parse tree!?



# Using Grammatical Structure

- Each constituent has *headword*.
- Condition on preceding *exposed* headwords?



# Using Grammatical Structure

- Predict next word based on preceding *exposed* headwords.

$P($	THE		▷	▷	)
$P($	DOG		▷	THE	)
$P($	ON		▷	DOG	)
$P($	TOP		DOG	ON	)
$P($	SAW		▷	DOG	)
$P($	ROY		DOG	SAW	)

- Picks most relevant preceding words ...
  - Regardless of position.
- Structured language model* (Chelba and Jelinek, 2000).

# Hey, Where Do Parse Trees Come From?

- Come up with grammar rules ...

S     →   NP VP  
NP    →   DET N | PN | NP PP  
N     →   dog | cat

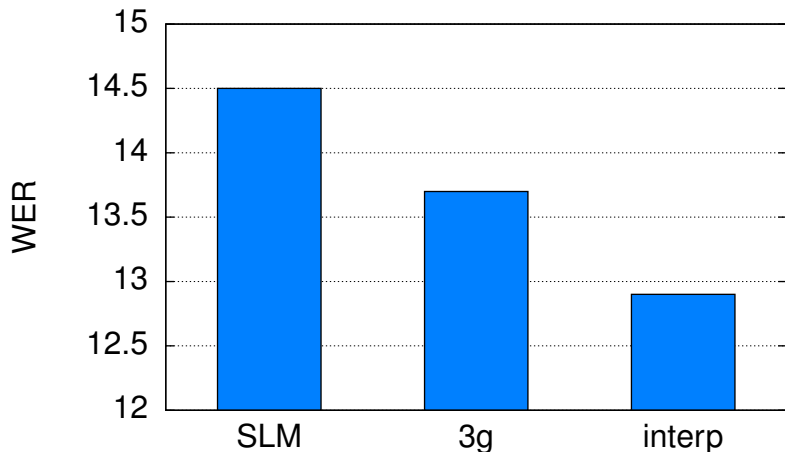
- Come up with probabilistic parametrization.

$$P_{\text{MLE}}(S \rightarrow \text{NP VP}) = \frac{c(S \rightarrow \text{NP VP})}{c(S)}$$

- Can extract rules and train probabilities using *treebank*.
  - *e.g.*, Penn Treebank (Switchboard, WSJ text).

# So, Does It Work?

- SLM trained on 20MW WSJ; trigram model: 40MW.



# Recap: Structured Language Modeling

- Grammatical language models not yet ready for prime time.
  - Need manually-parsed data to bootstrap parser.
  - Training/decoding is expensive, hard to implement.
- If have exotic LM and need publishable results . . .
  - Interpolate with trigram model (“ROVER effect”).

# Where Are We?

- 1 Short-Distance Dependencies: Word Classes
- 2 Medium-Distance Dependencies: Grammars
- 3 Long-Distance Dependencies: Adaptation
- 4 Decoding With Advanced Language Models
- 5 Discussion



# Modeling Long-Distance Dependencies

*A group including Phillip C. [Friedman](#) , a Gardena , California , investor , raised its stake in Genisco Technology Corporation to seven . five % of the common shares outstanding .*

*Neither officials of Compton , California - based Genisco , an electronics manufacturer , nor Mr. [Friedman](#) could be reached for comment .*

*In a Securities and Exchange Commission filing , the group said it bought thirty two thousand common shares between August twenty fourth and last Tuesday at four dollars and twenty five cents to five dollars each .*

*The group might buy more shares , its filing said .*

*According to the filing , a request by Mr. [Friedman](#) to be put on Genisco's board was rejected by directors .*

*Mr. [Friedman](#) has requested that the board delay Genisco's decision to sell its headquarters and consolidate several divisions until the decision can be " much more thoroughly examined to determine if it is in the company's interests , " the filing said .*

# Modeling Long-Distance Dependencies

- Observation: words and phrases in previous sentences ...
  - Are more likely to occur in future sentences.
- Language model *adaptation*.
  - Adapt language model to current style or topic.

$$P(w_i | w_{i-2} w_{i-1}) \Rightarrow P_{\text{adapt}}(w_i | w_{i-2} w_{i-1})$$

$$P(w_i | w_{i-2} w_{i-1}) \Rightarrow P(w_i | w_{i-2} w_{i-1}, H_i)$$

- Distribution over utterances  $P(\omega = w_1 \cdots w_l) \dots$ 
  - $\Rightarrow$  Utterance *sequences*  $P(\vec{\omega} = \omega_1 \cdots \omega_L)$ .

# Cache Language Models

- How to boost probabilities of recently-occurring words?
- Idea: build language model on last  $k = 500$  words, say.
- How to combine with primary language model?

$$P_{\text{cache}}(w_i | w_{i-2} w_{i-1}, w_{i-500}^{i-1}) = \\ \lambda \times P_{\text{static}}(w_i | w_{i-2} w_{i-1}) + (1 - \lambda) \times P_{w_{i-500}^{i-1}}(w_i | w_{i-2} w_{i-1})$$

- *Cache language models* (Kuhn and De Mori, 1990).

# Beyond Cache Language Models

- What's the problem?
  - Does seeing THE boost the probability of THE?
  - Does seeing MATSUI boost the probability of YANKEES?
- Can we induce which words *trigger* which other words?
  - How might one find trigger pairs?

HENSON	MUPPETS
TELESCOPE	ASTRONOMERS
CLOTS	DISSOLVER
NODES	LYMPH
SPINKS	HEAVYWEIGHT
DYSTROPHY	MUSCULAR
FEEDLOTS	FEEDLOT
SCHWEPPE	MOTT'S

# Trigger Language Models

- How to combine with primary language model?
  - Linear interpolation with trigger unigram?

$$P_{\text{trig}}(w_i | w_{i-2} w_{i-1}, w_{i-500}^{i-1}) = \\ \lambda \times P_{\text{static}}(w_i | w_{i-2} w_{i-1}) + (1 - \lambda) \times P_{w_{i-500}^{i-1}}(w_i)$$

- Another way: *maximum entropy* models (Lau *et al.*, 1993).

# Beyond Trigger Language Models

- Some groups of words are mutual triggers.
  - *e.g.*, IMMUNE, LIVER, TISSUE, TRANSPLANTS, etc.
  - Difficult to discover all pairwise relations: sparse.
- May not want to trigger words based on single event.
  - Some words are ambiguous.
  - *e.g.*, LIVER  $\Rightarrow$  TRANSPLANTS or CHICKEN?
- $\Rightarrow$  Topic language models.

# Example: Seymore and Rosenfeld (1997)

- Assign topics to documents.
  - *e.g.*, politics, medicine, Monica Lewinsky, cooking, etc.
  - Manual labels (*e.g.*, BN) or unsupervised clustering.
- For each topic, build topic-specific LM.
- Decoding.
  - 1st pass: use generic LM.
  - Select topic LM's maximizing likelihood of 1st pass.
  - Re-decode using topic LM's.

# Example: Seymore and Rosenfeld (1997)

- Training (transcript); topics: conspiracy; JFK assassination.

THEY WERE RIDING THROUGH DALLAS WITH THE KENNEDYS  
WHEN THE FAMOUS SHOTS WERE FIRED

HE WAS GRAVELY WOUNDED

HEAR WHAT GOVERNOR AND MRS. JOHN CONNALLY THINK OF  
THE CONSPIRACY MOVIE J. F. K. ...

- Test (decoded); topics: ???

THE MURDER OF J. F. K. WAS IT A CONSPIRACY  
SHOULD SECRET GOVERNMENT FILES BE OPENED TO THE  
PUBLIC

CAN THE TRAGIC MYSTERY EVER BE SATISFACTORILY  
RESOLVED ...



# Example: Seymore and Rosenfeld (1997)

- Topic LM's may be sparse.
  - Combine with general LM.
- How to combine selected topic LM's and general LM?
  - Linear interpolation!

$$P_{\text{topic}}(w_i | w_{i-2} w_{i-1}) = \lambda_0 P_{\text{general}}(w_i | w_{i-2} w_{i-1}) + \sum_{t=1}^T \lambda_t P_t(w_i | w_{i-2} w_{i-1})$$

# So, Do Cache Models Work?

- Um, -cough-, kind of.
- Good PP gains (up to  $\sim 20\%$ ).
- WER gains: little to none.
  - *e.g.*, (Iyer and Ostendorf, 1999; Goodman, 2001).

# What About Trigger and Topic Models?

- Triggers.
  - Good PP gains (up to  $\sim 30\%$ )
  - WER gains: unclear; *e.g.*, (Rosenfeld, 1996).
- Topic models.
  - Good PP gains (up to  $\sim 30\%$ )
  - WER gains: up to 1% absolute.
  - *e.g.*, (Iyer and Ostendorf, 1999; Goodman, 2001).

# Recap: Adaptive Language Modeling

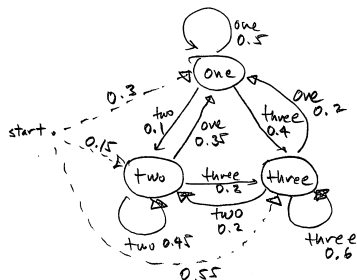
- ASR errors can cause adaptation errors.
  - In lower WER domains, LM adaptation may help more.
- Large PP gains, but small WER gains.
  - What's the dillio?
- Increases system complexity for ASR.
  - *e.g.*, how to adapt LM scores with static decoding?
- Unclear whether worth the effort.

# Where Are We?

- 1 Short-Distance Dependencies: Word Classes
- 2 Medium-Distance Dependencies: Grammars
- 3 Long-Distance Dependencies: Adaptation
- 4 **Decoding With Advanced Language Models**
- 5 Discussion

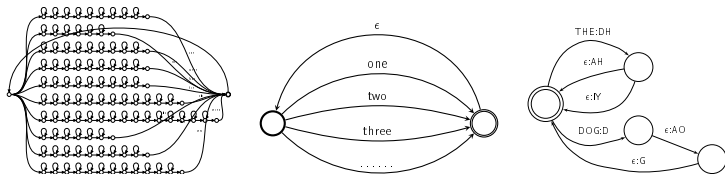
# Decoding, Class $N$ -Gram Models

- Can we build the one big HMM?
- Start with class  $n$ -gram model as FSA.
- Expand each class to *all* members.



# Handling Complex Language Models

- One big HMM: *static* graph expansion.
  - Heavily-pruned  $n$ -gram language model.
- Another approach: *dynamic* graph expansion.
  - Don't store whole graph in memory.
  - Build parts of graph with active states on the fly.



# Dynamic Graph Expansion: The Basic Idea

- Express graph as composition of two smaller graphs.
  - Composition is associative.

$$\begin{aligned}G_{\text{decode}} &= L \circ T_{\text{LM} \rightarrow \text{CI}} \circ T_{\text{CI} \rightarrow \text{CD}} \circ T_{\text{CD} \rightarrow \text{GMM}} \\ &= L \circ (T_{\text{LM} \rightarrow \text{CI}} \circ T_{\text{CI} \rightarrow \text{CD}} \circ T_{\text{CD} \rightarrow \text{GMM}})\end{aligned}$$

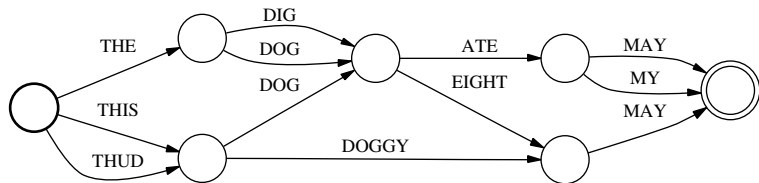
- Can do *on-the-fly* composition.
  - States in result correspond to state pairs  $(s_1, s_2)$ .



# Another Way: Two-Pass Decoding

- First-pass decoding: use simpler model ...
  - To find “likeliest” word *sequences* ...
  - As lattice (WFSA) or flat list of hypotheses ( $N$ -best list).
- *Rescoring*: use complex model ...
  - To find best word sequence in lattice/list.

# Lattice Generation and Rescoring



- In Viterbi, store  $k$ -best tracebacks at each word-end cell.
- To add in new LM scores to lattice ...
  - What operation can we use?

# *N*-Best List Rescoring

- For exotic models, even lattice rescoring may be too slow.
- Easy to generate *N*-best lists from lattices ( $A^*$  algorithm).

THE DOG ATE MY  
THE DIG ATE MY  
THE DOG EIGHT MAY  
THE DOGGY MAY

# Discussion: A Tale of Two Decoding Styles

- Approach 1: Dynamic graph expansion (since late 1980's).
  - Can handle more complex language models.
  - Decoders are incredibly complex beasts.
  - *e.g.*, cross-word CD expansion without FST's.
  - Graph optimization difficult.
- Approach 2: Static graph expansion (AT&T, late 1990's).
  - Enabled by optimization algorithms for WFSM's.
  - Much cleaner way of looking at everything!
  - FSM toolkits/libraries can do a lot of work for you.
  - Static graph expansion is complex, but offline!
  - Decoding is relatively simple.

# Static or Dynamic? Two-Pass?

- If speed is priority?
- If flexibility is priority?
  - *e.g.*, update LM vocabulary every night.
- If need gigantic language model?
- If latency is priority?
  - What can't we use?
- If accuracy is priority (all the time in the world)?
- If doing cutting-edge research?

# Where Are We?

- 1 Short-Distance Dependencies: Word Classes
- 2 Medium-Distance Dependencies: Grammars
- 3 Long-Distance Dependencies: Adaptation
- 4 Decoding With Advanced Language Models
- 5 Discussion

# Recap

- Short-distance dependencies.
  - Interpolate class  $n$ -gram with word  $n$ -gram.
  - $<1\%$  absolute WER gain; pain to implement?
- Medium-distance dependencies.
  - Interpolate grammatical LM with word  $n$ -gram.
  - $<1\%$  absolute WER gain; pain to implement.
- Long-distance dependencies.
  - Interpolate adaptive LM with static  $n$ -gram.
  - $<1\%$  absolute WER gain; pain to implement.
- PP  $\neq$  WER.

# Turning It Up To Eleven (Goodman, 2001)

- If short, medium, and long-distance modeling ...
  - All achieve  $\sim 1\%$  WER gain ...
  - What if combine them all with linear interpolation?
- “A Bit of Progress in Language Modeling”.
  - Combined higher order  $n$ -grams, skip  $n$ -grams, ...
  - Class  $n$ -grams, cache models, sentence mixtures.
  - Achieved 50% reduction in PP over word trigram.
  - $\Rightarrow \sim 1\%$  WER gain (WSJ  $N$ -best list rescoring).



# State of the Art Circa 2005





- Commercial systems.
  - Word  $n$ -gram models.
- Research systems, *e.g.*, government evaluations.
  - No time limits; tiny differences in WER matter.
  - Interpolation of word 4-gram models.
- Why aren't people using ideas from LM research?
  - Too slow (1st pass decoding; rescoring?)
  - Gains not reproducible with largest data sets.

# Time To Give Up?





*... we argue that meaningful, practical reductions in word error rate are hopeless. We point out that trigrams remain the de facto standard not because we don't know how to beat them, but because no improvements justify the cost.*

*— Joshua Goodman (2001)*

# References

-  P.F. Brown, V.J. Della Pietra, P.V. deSouza, J.C. Lai, R.L. Mercer, "Class-based n-gram models of natural language", Computational Linguistics, vol. 18, no. 4, pp. 467–479, 1992.
-  C. Chelba and F. Jelinek, "Structured language modeling", Computer Speech and Language, vol. 14, pp. 283–332, 2000.
-  J.T. Goodman, "A Bit of Progress in Language Modeling", Microsoft Research technical report MSR-TR-2001-72, 2001.
-  R. Iyer and M. Ostendorf, "Modeling Long Distance Dependence in Language: Topic Mixtures vs. Dynamic Cache Models", IEEE Transactions on Speech and Audio Processing, vol. 7, no. 1, pp. 30–39, 1999.

# References (cont'd)

-  R. Kuhn and R. De Mori, “A Cache-Based Natural Language Model for Speech Reproduction”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 6, pp. 570–583, 1990.
-  R. Lau, R. Rosenfeld, S. Roukos, “Trigger-based Language Models: A Maximum Entropy Approach”, ICASSP, vol. 2, pp. 45–48, 1993.
-  R. Rosenfeld, “A Maximum Entropy Approach to Adaptive Statistical Language Modeling”, Computer Speech and Language, vol. 10, pp. 187–228, 1996.
-  K. Seymore and R. Rosenfeld, “Using Story Topics for Language Model Adaptation”, Eurospeech, 1997.

## Part II

# Language Modeling, Post-2005-ish

# What Up?

- Humans use short, medium, and long-distance info.
- Sources of info seem complementary.
- Yet, linear interpolation fails to yield cumulative gains.
- View: interpolation  $\Leftrightarrow$  averaging; *dilutes* each model.
- Maybe instead of hammer, need screwdriver?

# Is There Another Way?

- Can we combine multiple information sources ...
- Such that resulting language model ...
- Enforces each information source **fully**!?

# Where Are We?

- 1 Introduction to Maximum Entropy Modeling
- 2 *N*-Gram Models and Smoothing, Revisited
- 3 Maximum Entropy Models, Part III
- 4 Neural Net Language Models
- 5 Discussion



# Many Models Or One?

- Old view.
  - Build one model for each information source.
  - Interpolate.
- New view.
  - Build one model.

# Types of Information To Combine

- Word  $n$ -gram.
- Class  $n$ -gram.
- Grammatical information.
- Cache, triggers.
- Topic information.
- Is there a common way to express all this?

# The Basic Intuition

- Say we have 1M utterances of training data  $\mathcal{D}$ .

FEDERAL HOME LOAN MORTGAGE CORPORATION –DASH ONE  
.POINT FIVE BILLION DOLLARS OF REALESTATE MORTGAGE  
-HYPHEN INVESTMENT CONDUIT SECURITIES OFFERED BY  
MERRILL LYNCH &AMPERSAND COMPANY .PERIOD

NONCOMPETITIVE TENDERS MUST BE RECEIVED BY NOON  
EASTERN TIME THURSDAY AT THE TREASURY OR AT FEDERAL  
RESERVE BANKS OR BRANCHES .PERIOD ...

- Train LM  $P(\omega)$  on  $\mathcal{D}$ ; generate 1M utterances  $\mathcal{D}'$ .
- If THE occurs  $1.062 \times 10^6$  times in  $\mathcal{D}$  ...
- How many times should occur in  $\mathcal{D}'$ ?

# Marginals

- Frequency of THE in  $\mathcal{D}$  and  $\mathcal{D}'$  should match:

$$\begin{aligned}c_{\mathcal{D}}(\text{THE}) &= c_{\mathcal{D}'}(\text{THE}) \\&= N \sum_{h,w:hw=\dots\text{THE}} P(h, w) \\&= N \sum_{h,w:hw=\dots\text{THE}} c_{\mathcal{D}}(h)P(w|h)\end{aligned}$$

- Bigram:  $hw = \dots$  OF THE.
- Class bigram:  $hw$  ends in classes [FOOD] [PREP].
- Grammar:  $hw$  is prefix of grammatical sentence.
- Trigger: HENSON in last 500 words of  $h$  and  $w =$  MUPPETS.

# Marginal Constraints

- Binary *feature* functions, e.g.,  $hw = \dots \text{THE}$

$$f_{\text{THE}}(h, w) = \begin{cases} 1 & \text{if } hw = \dots \text{THE} \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} c_{\mathcal{D}}(\text{THE}) &= N \sum_{h, w: hw = \dots \text{THE}} c_{\mathcal{D}}(h) P(w|h) \\ &= N \sum_{h, w} c_{\mathcal{D}}(h) P(w|h) f_{\text{THE}}(h, w) \end{aligned}$$

- Select model  $P(w|h)$  satisfying all marginals.
- Which one!?

# Maximum Entropy Principle (Jaynes, 1957)

- The entropy  $H(P)$  of  $P(w|h)$  is

$$H(P) = - \sum_{\omega} P(h, w) \log P(w|h)$$

- Entropy  $\Leftrightarrow$  uniformness  $\Leftrightarrow$  least assumptions.
- Of models satisfying constraints ...
  - Pick one with highest entropy!
  - Capture constraints; assume nothing more!

# Can We Find the Maximum Entropy Model?

- Features  $f_1(x, y), \dots, f_F(x, y)$ ; parameters  $\Lambda = \{\lambda_1, \dots, \lambda_F\}$ .
- ME model satisfying associated constraints has form:

$$P_{\Lambda}(w|h) = \frac{1}{Z_{\Lambda}(h)} \prod_{i: f_i(h, w)=1} e^{\lambda_i}$$

$$\log P_{\Lambda}(w|h) = \sum_{i=1}^F \lambda_i f_i(h, w) - \log Z_{\Lambda}(h)$$

- $Z_{\Lambda}(h)$  = normalizer =  $\sum_w \exp(\sum_{i=1}^F \lambda_i f_i(h, w))$ .
- a.k.a. *exponential model*, *log-linear model*.

# How to Find the $\lambda_i$ 's?

$$P_{\Lambda}(w|h) = \frac{1}{Z_{\Lambda}(h)} \prod_{i: f_i(h,w)=1} e^{\lambda_i}$$

- $\{\lambda_i\}$ 's satisfying constraints are MLE's!
- Training set likelihood is convex function of  $\{\lambda_i\}$ !



# Recap: Maximum Entropy Modeling

- Elegant as all hell.
- Principled way to combine lots of information sources.
  - Design choice: which constraints to enforce?
- Single global optimum when training parameters.
- Interpolation: addition; ME: multiplication.
- But does it blend?

# Where Are We?

- 1 Introduction to Maximum Entropy Modeling
- 2 **N-Gram Models and Smoothing, Revisited**
- 3 Maximum Entropy Models, Part III
- 4 Neural Net Language Models
- 5 Discussion

# Maximum Entropy $N$ -gram Models?

- Can ME help build a better word  $n$ -gram model?
- One constraint per seen  $n$ -gram.

$$f_{\text{I LIKE BIG}}(h, w) = \begin{cases} 1 & \text{if } hw = \dots \text{I LIKE BIG} \\ 0 & \text{otherwise} \end{cases}$$

- Problem: MLE model is same as before!!!

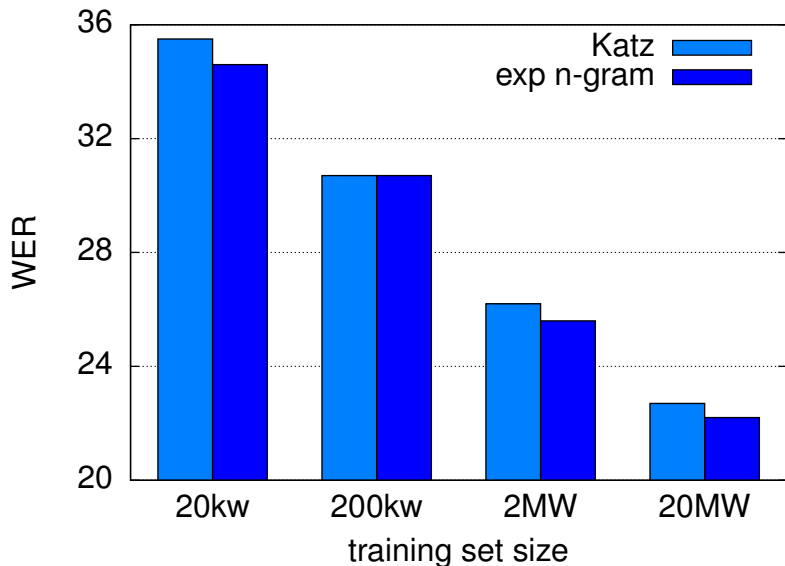
# Smoothing for Exponential Models

- Point: don't want to match training counts *exactly*!
- e.g.,  $\ell_2^2$  regularization (e.g., Chen and Rosenfeld, 2000).

$$\text{obj fn} = \text{LL}_{\text{train}} + \frac{1}{(\# \text{ train wds})} \sum_{i=1}^F \frac{\lambda_i^2}{2\sigma^2}$$

- The smaller  $|\lambda_i|$  is, the smaller its effect ...
  - And the smoother the model.

# Smoothing for Exponential Models (WSJ 4g)



# Yay!?

- Smoothed exponential  $n$ -gram models perform well.
- Why don't people use them?
  - Conventional  $n$ -gram: count and normalize.
  - Exponential  $n$ -gram: 50 rounds of iterative scaling.
- Is there way to do constraint-based modeling ...
  - Within conventional  $n$ -gram framework?

# Kneser-Ney Smoothing (1995)

- Back-off smoothing.

$$P_{\text{KN}}(w_i | w_{i-1}) = \begin{cases} P_{\text{primary}}(w_i | w_{i-1}) & \text{if } c(w_{i-1} w_i) > 0 \\ \alpha_{w_{i-1}} P_{\text{KN}}(w_i) & \text{otherwise} \end{cases}$$

- $P_{\text{KN}}(w_i)$  chosen such that ...
  - Unigram constraints met exactly.

# Kneser-Ney Smoothing

- Unigram probabilities  $P_{\text{KN}}(w_i) \dots$
- *Not* proportional to how often unigram occurs.

$$P_{\text{KN}}(w_i) \neq \frac{c(w_i)}{\sum_{w_i} c(w_i)}$$

- Proportional to how many word types unigram follows!

$$N_{1+}(\bullet w_i) \equiv |\{w_{i-1} : c(w_{i-1} w_i) > 0\}|$$

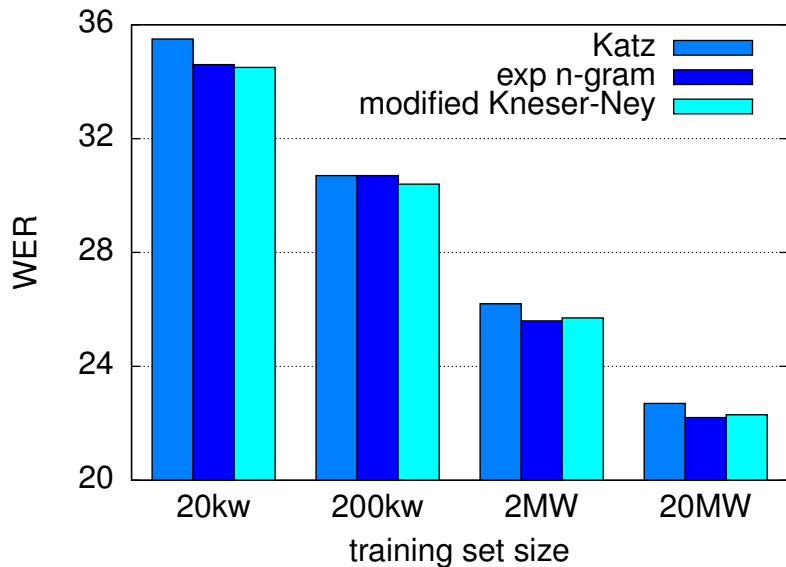
$$P_{\text{KN}}(w_i) = \frac{N_{1+}(\bullet w_i)}{\sum_{w_i} N_{1+}(\bullet w_i)}$$

---

<sup>†</sup> Check out “A Hierarchical Bayesian Language Model based on Pitman-Yor Processes”, Teh, 2006, for a cool Bayesian interpretation and learn about Chinese restaurant processes.



# Kneser-Ney Smoothing



# Recap: $N$ -Gram Models and Smoothing

- Best  $n$ -gram smoothing methods are all constraint-based.
- Can express smoothed  $n$ -gram models as ...
  - Exponential models with simple  $\ell_2^2$  smoothing.
- “Modified” interpolated Kneser-Ney smoothing<sup>†</sup> ...
  - Yields similar model, but much faster training.
  - Standard in literature for last 15+ years.
- Available in SRI LM toolkit.

<http://www.speech.sri.com/projects/srilm/>

---

<sup>†</sup>(Chen and Goodman, 1998).

# Where Are We?

- 1 Introduction to Maximum Entropy Modeling
- 2 *N*-Gram Models and Smoothing, Revisited
- 3 Maximum Entropy Models, Part III**
- 4 Neural Net Language Models
- 5 Discussion

# What About Other Features?

- Exponential models make slightly better  $n$ -gram models.
  - Snore.
- Can we just toss in tons of cool features ...
  - And get fabulous results?

# Maybe!? (Rosenfeld, 1996)

- 38M words of WSJ training data.
- Trained maximum entropy model with ...
  - Word  $n$ -gram; skip  $n$ -gram; trigger features.
  - Interpolated with regular word  $n$ -gram and cache.
- 39% reduction in PP, 2% absolute reduction in WER.
  - Baseline: (pruned) Katz-smoothed(?) trigram model.
- Contrast: Goodman (2001), -50% PP, -0.9% WER.

# What's the Catch?

- 200 computer-days to train.
- Really slow training.
  - For each word, update  $O(|V|)$  counts.
  - Tens of passes through training data.
- Really slow evaluation: evaluating  $Z_{\Lambda}(h)$ .

$$P_{\Lambda}(w|h) = \frac{1}{Z_{\Lambda}(h)} \prod_{i:f_i(h,w)=1} e^{\lambda_i}$$

$$Z_{\Lambda}(h) = \sum_{w'} \exp\left(\sum_{i=1}^F \lambda_i f_i(h, w')\right)$$

# Newer Developments

- Fast training: optimizations for simple feature sets.
  - *e.g.*, train word  $n$ -gram model on 1GW in few hours.
- Fast evaluation: unnormalized models.
  - Not much slower than regular word  $n$ -gram.

$$P_{\Lambda}(w|h) = \prod_{i:f_i(h,w)=1} e^{\lambda_i}$$

- Performance prediction.
  - How to intelligently select feature types.

# Performance Prediction (Chen, 2008)

- Given training set and test set from same distribution.
- Desire: want to optimize performance on *test* set.
- Reality: only have access to *training* set.

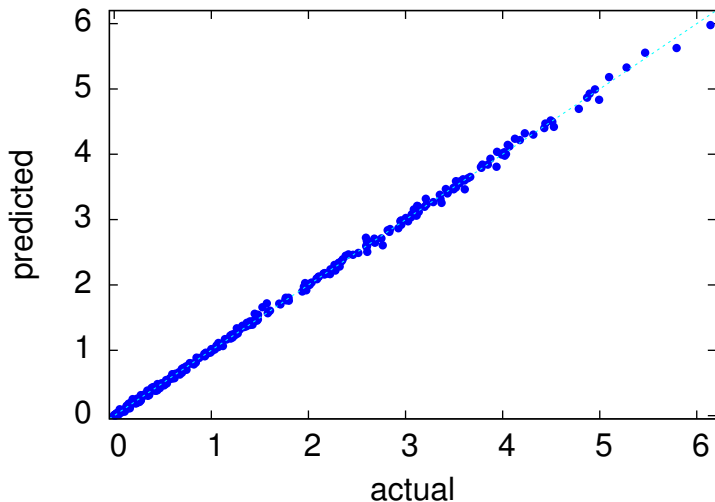
$$(\text{test perf}) = (\text{training perf}) + (\text{overfitting penalty})$$

- Can we estimate *overfitting penalty*?



# Yes

$$\log PP_{\text{test}} - \log PP_{\text{train}} \approx \frac{0.938}{(\# \text{ train wds})} \sum_{i=1}^F |\lambda_i|$$



# A Tool for Good

- Holds for many different types of data.
  - Different domains; languages; token types; ...
  - Vocab sizes; training set sizes;  $n$ -gram orders.
- Holds for many different types of exponential models.
- Explains lots of diverse aspects of language modeling.
- Can choose features types ...
  - To *intentionally* shrink  $\sum_{i=1}^F |\lambda_i|$ .

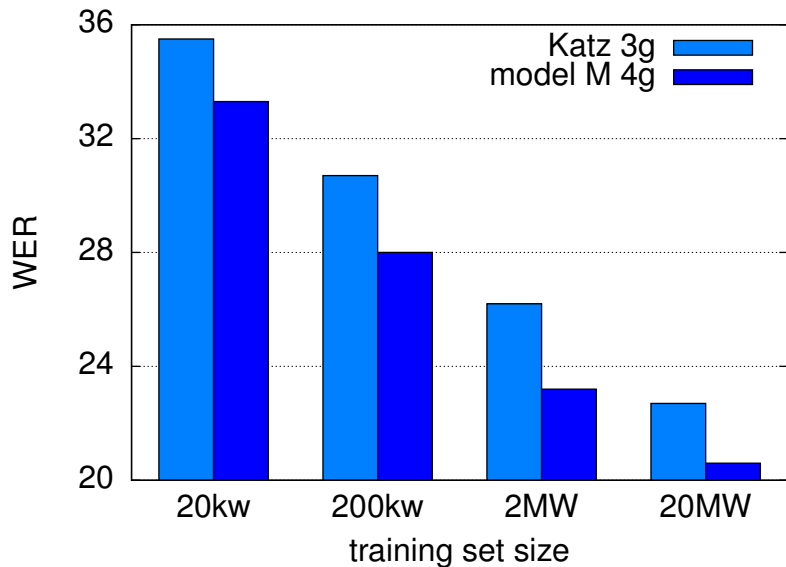
# Model M (Chen, 2008; Chen and Chu, 2010)

- Old-timey class-based model (Brown, 1992).
  - Class prediction features:  $c_{i-2}c_{i-1}c_i$ .
  - Word prediction features:  $c_iw_i$ .

$$P(w_i|w_{i-2}w_{i-1}) = P(c_i|c_{i-2}c_{i-1}) \times P(w_i|c_i)$$

- Start from word  $n$ -gram model; convert to class model ...
  - And choose feature types to reduce overfitting.
  - Class prediction features:  $c_{i-2}c_{i-1}c_i, w_{i-2}w_{i-1}c_i$ .
  - Word prediction features:  $w_{i-2}w_{i-1}c_iw_i$ .
- Without interpolation with word  $n$ -gram model.

# Model M (WSJ)



# Recap: Maximum Entropy

- Some of best WER results in LM literature.
  - Gain of up to 3% absolute WER over trigram (not  $<1\%$ ).
  - Short-range dependencies only.
- Can surpass linear interpolation in WER in many contexts.
  - *Log*-linear interpolation.
  - Each is appropriate in different situations. (When?)
  - Together, powerful tool set for model combination.
- Performance prediction explains existing models ...
  - And helps design new ones!
- Training can be painful depending on features.

# Where Are We?

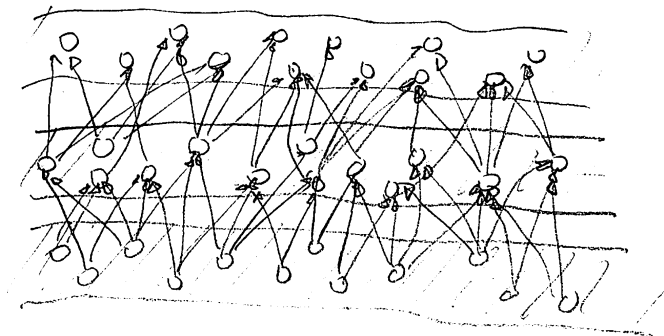
- 1 Introduction to Maximum Entropy Modeling
- 2 *N*-Gram Models and Smoothing, Revisited
- 3 Maximum Entropy Models, Part III
- 4 Neural Net Language Models**
- 5 Discussion

# Introduction

- Ways to combine information sources.
  - Linear interpolation.
  - Exponential/log-linear models.
  - Anything else?
- Recently, good results with *neural networks*.

# What is the Brain Like?

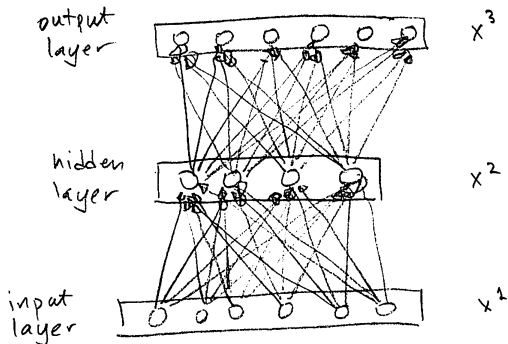
- Layered; mostly “forward” connections.
- Each neuron has *firing rate*.





# What is a Basic Neural Network Like?

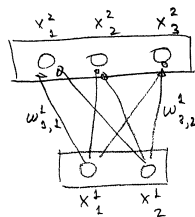
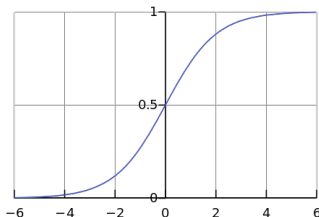
- Layered; only “forward” connections and full.
- Each neuron has *activation*.



# How Are Activations Computed?

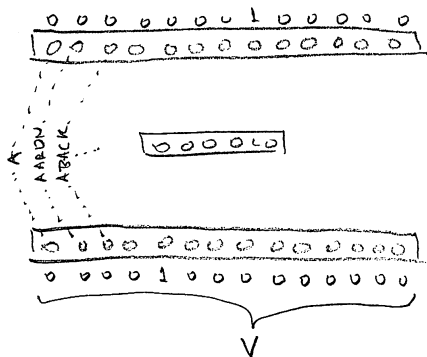
- Linear function of previous layer ...
- Then, non-linearity  $\Rightarrow [0, 1]$  (maybe).
  - e.g., sigmoid:  $g(x) = \frac{1}{1+e^{-x}}$ .
  - Saturation; universal function approximator.

$$x_i^{l+1} = g\left(\sum_{j=1}^{N_l} w_{ij}^l x_j^l\right)$$



# Encoding Inputs and Outputs

- e.g., how to model  $P(w_i|w_{i-1})$ ?
- 1 of  $V$  coding:  $V$  nodes, one on and the rest off.



# Probability Estimation

- How to make final layer activations act like probs?
- Use *softmax* function.

$$p_i = \frac{e^{x_i^L}}{\sum_{j=1}^{N_L} e^{x_j^L}}$$

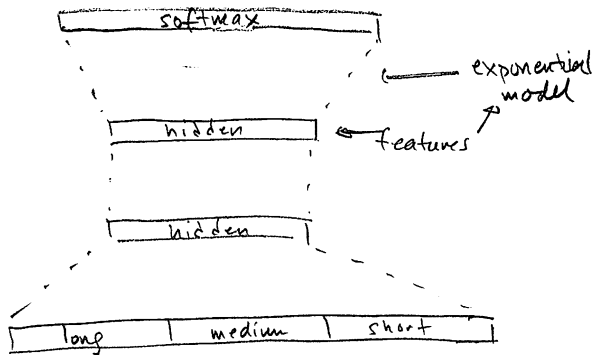
- Use log likelihood as training objective function.
  - a.k.a. *cross-entropy*.

# What's the Big Deal?

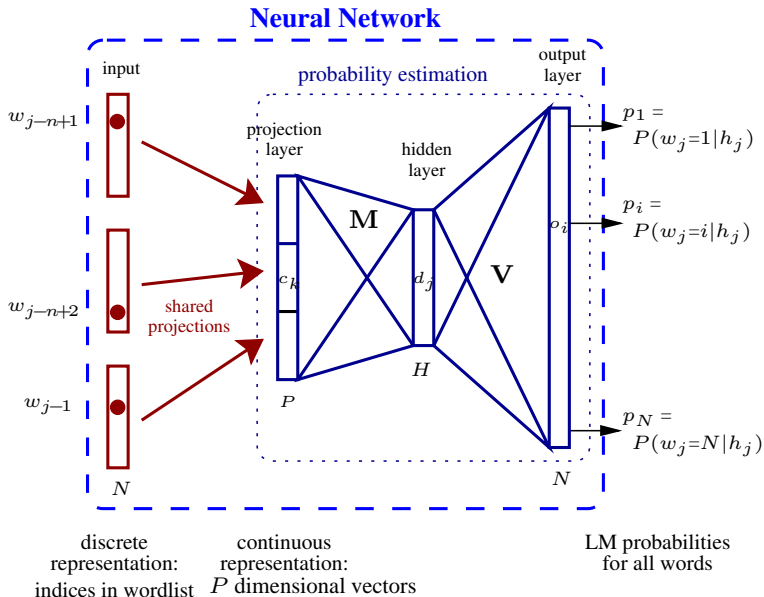
- Multi-layer models are better than single-layer models.
- Can actually *train* them scalably!
  - *Backpropagation*  $\Rightarrow$  gradient-descent.
  - Unlike graphical models.
- Computers are fast enough now, *e.g.*, GPU's.

# Perspective: Exponential Models

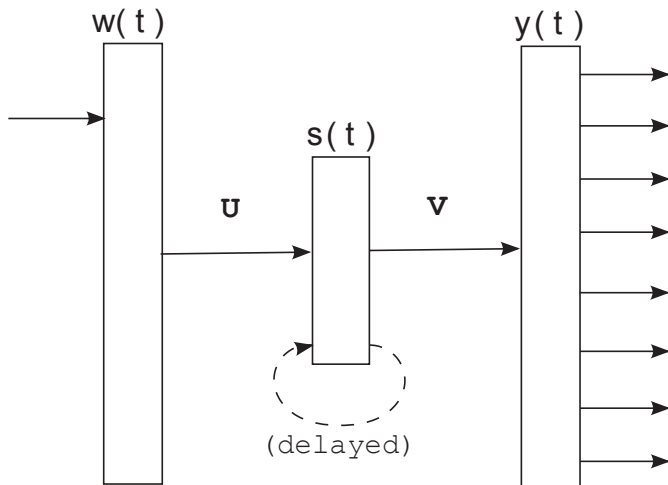
- Last layer: log-linear + softmax  $\Rightarrow$  maxent!
- Like exponential/ME model, but *learn* features!



# NNLM's 1.0 (Schwenk and Gauvain, 2005)



# Recurrent NN LM's (Mikolov *et al.*, 2010)





# Results (Mikolov *et al.*, 2011)

Model	Dev WER[%]	Eval WER[%]
Baseline - KN5	12.2	17.2
Discriminative LM [14]	11.5	16.9
Joint LM [7]	-	16.7
Static RNN	10.5	14.9
Static RNN + KN	10.2	14.6
Adapted RNN	9.8	14.5
Adapted RNN + KN	9.8	14.5
All RNN	<b>9.7</b>	<b>14.4</b>

# Discussion

- Some of best WER results in LM literature.
  - Gain of up to 3% absolute WER over trigram (not <1%).
- Interpolation with word  $n$ -gram optional.
- Can integrate arbitrary features, *e.g.*, syntactic features.
  - Easy to condition on longer histories.
- Training and evaluation is slow.
  - Optimizations: class-based modeling; reduced vocab.
- Publicly-available toolkit: <http://rnnlm.org>

# Where Are We?

- 1 Introduction to Maximum Entropy Modeling
- 2 *N*-Gram Models and Smoothing, Revisited
- 3 Maximum Entropy Models, Part III
- 4 Neural Net Language Models
- 5 Discussion**

# Other Directions in Language Modeling

Discriminative training for LM's.

Super ARV LM.

LSA-based LM's.

Variable-length  $n$ -grams; skip  $n$ -grams.

Concatenating words to use in classing.

Context-dependent word classing.

Word classing at multiple granularities.

Alternate parametrizations of class  $n$ -grams.

Using part-of-speech tags.

Semantic structured LM.

Sentence-level mixtures.

Soft classing.

Hierarchical topic models.

Combining data/models from multiple domains.

Whole-sentence maximum entropy models.

# State of the Art, Production Systems

- Word  $n$ -gram models (1st pass static).
  - Modified Kneser-Ney smoothing?
- Mostly word  $n$ -gram models (dynamic/rescoring).
- Gain not worth the cost/complexity.
  - The more data, the less gain!

	train FLOPS/ev	eval FLOPS/ev
$n$ -gram	5	1–3
model M	$20 \times 100$	5–10
NNLM	$10^6 \times 20$	$10^6$

# State of the Art, Research Systems

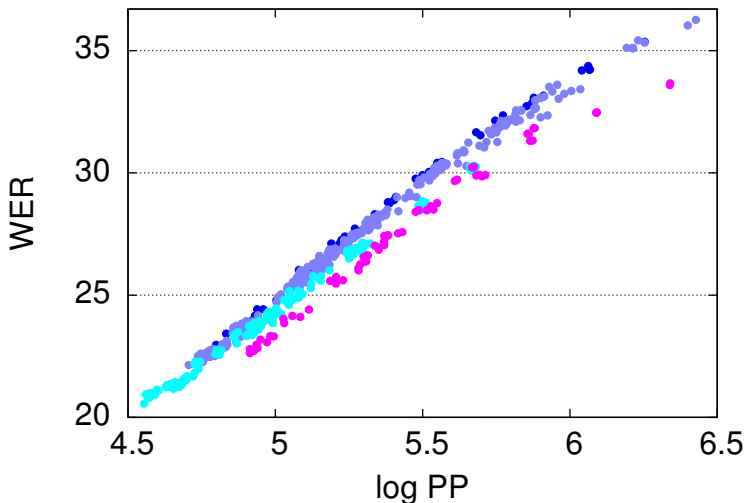
- e.g., government evals.
  - Small differences in WER matter; may not have much data.
  - Interpolation of word  $n$ -gram models.
  - Rescoring w/ neural net LM's; Model M (-0.5% WER?)
- Modeling medium-to-long-distance dependencies.
  - Almost no gain in combination with other techniques?
  - Not worth extra effort and complexity.

# An Apology to $N$ -Gram Models

- I didn't mean what I said about you.
- You know I was kidding when I said ...
  - You are great to poop on.

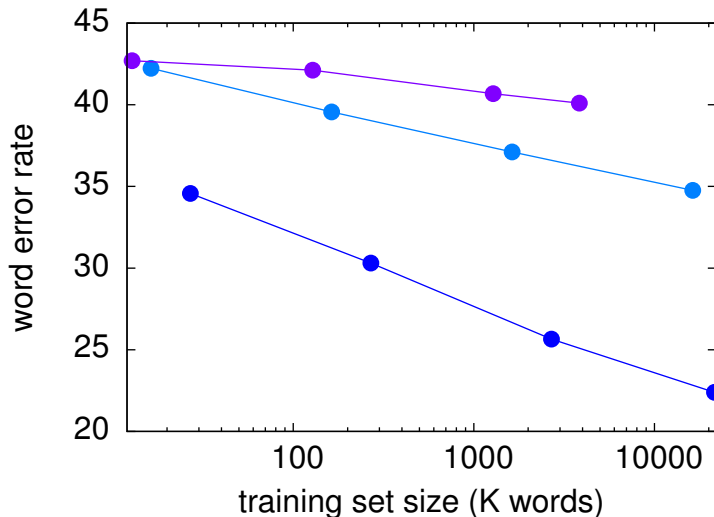
# Lessons: Perplexity $\neq$ WER

- e.g., One Billion Word Benchmark.
- Vast perplexity improvements  $\Rightarrow$  small WER gains.

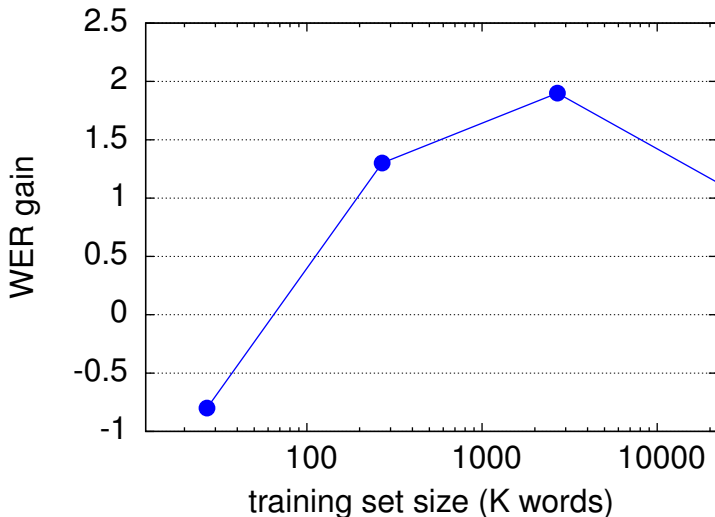




# What's Important: Data!








# What's Not As Important: Algorithms!








# Where Do We Go From Here?

- $N$ -gram models are just really easy to build.
  - Smarter LM's tend to be orders of magnitude slower.
  - Faster computers? Data sets also growing.
- Need to effectively combine many sources of information.
  - Short, medium, and long distance.
  - Log-linear models, NN's promising, but slow to train.
- Evidence that LM's will help more when WER's are lower.
  - Human rescoring of  $N$ -best lists (Brill *et al.*, 1998).

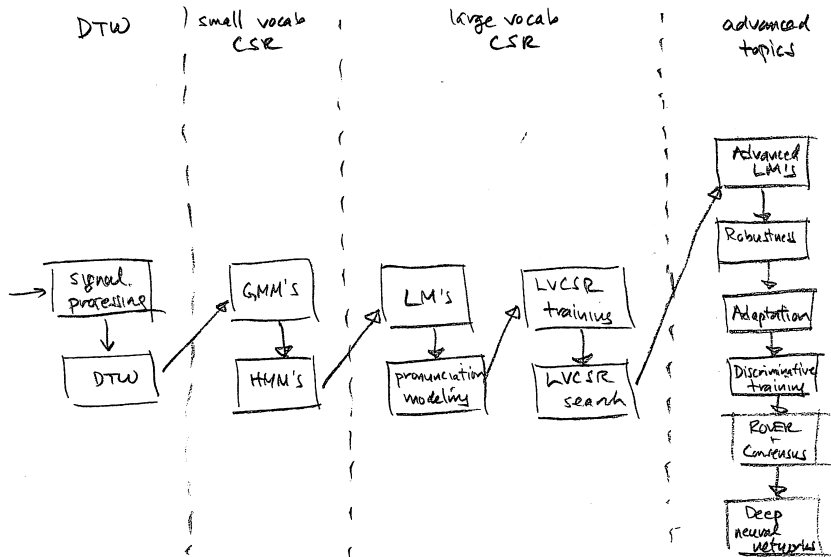
# References

-  A.L. Berger, S.A. Della Pietra, V.J. Della Pietra, “A maximum entropy approach to natural language processing”, Computational Linguistics, vol. 22, no. 1, pp. 39–71, 1996.
-  E. Brill, R. Florian, J.C. Henderson, L. Mangu, “Beyond N-Grams: Can Linguistic Sophistication Improve Language Modeling?”, ACL, pp. 186–190, 1998.
-  S.F. Chen, “Performance Prediction for Exponential Language Models”, IBM Research Division technical report RC 24671, 2008.
-  S.F. Chen and S.M. Chu, “Enhanced Word Classing for Model M”, Interspeech, 2010.
-  S.F. Chen and J. Goodman, “An Empirical Study of Smoothing Techniques for Language Modeling”, Harvard University technical report TR-10-98, 1998.

# References

-  S.F. Chen and R. Rosenfeld, “A Survey of Smoothing Techniques for Maximum Entropy Models”, IEEE Transactions on Speech and Audio Processing, vol. 8, no. 1, pp. 37–50, 2000.
-  E.T. Jaynes, “Information Theory and Statistical Mechanics”, Physics Reviews, vol. 106, pp. 620–630, 1957.
-  R. Kneser and H. Ney, “Improved Backing-off for M-Gram Language Modeling”, ICASSP, vol. 1, pp. 181–184, 1995.
-  H. Schwenk and J.L. Gauvain, “Training Neural Network Language Models on Very Large Corpora”, HLT/EMNLP, pp. 201–208, 2005.
-  T. Mikolov, “Statistical Language Models based on Neural Networks”, Ph.D. thesis, Brno University of Technology, 2012.

# Road Map



# Course Feedback

- 1 Was this lecture mostly clear or unclear? What was the muddiest topic?
- 2 Other feedback (pace, content, atmosphere)?