Adaptive Statistical Language Modeling: A Maximum Entropy Approach

Ronald Rosenfeld

School of Computer Science

Carnegie Mellon

Outline

1. Language Modeling
2. Why Statistical Language Modeling?
3. Fundamental Issues
4. Information Sources
5. Combining Information Sources
6. The Maximum Entropy Principle
7. Using Maximum Entropy in Language Modeling
8. Adaptation in Language Modeling
9. Contributions / Significance
10. Future Directions / Research Agenda

Language Modeling

"The attempt to characterize, capture and exploit regularities in natural language."

Applications:

- Automatic Speech Recognition
- Machine Translation
- Optical Character Recognition
- Spelling Correction

Why is it important *now*?

- Involved in 3 of the 4 SILK technologies.
- Crucial effect on performance; much room for improvement.
- Lots of data made available recently.

Why Statistical Language Modeling?

1. Why not hand-code theories of language?

- Experts are often right in principle but wrong in detail. Data is the best expert on details (lesson from acoustic modeling).
- Our data (and ability to handle it) grow faster than our linguistic understanding.

2. Language has a deep structure. Why not use it?

- Current parsing technology is inadequate.
- Traditional parsers answer 'yes'/'no'.
- Parsers typically handle only grammatical language.
- Parsers are slow.

Fundamental Issues
[& State-of-the-art Circa 1992]

1. Identifying knowledge sources
   [Short-term only; trigram (1975) undefeated]

2. Combining knowledge sources
   [Linear interpolation (1975)]

3. Adaptation (domains, speaking styles, time shift)
   [Simple Ngram caches (Kuhn 1988, IBM 1991)]

4. Exploiting the deep structure of language
   [None]

An Information Theory Perspective

1. Fundamental equation of speech recognition:

$$\arg\max_{L} \Pr(L|A) \;=\; \arg\max_{L} \frac{\Pr(A|L) \cdot \Pr(L)}{\Pr(A)}$$

2. The Chain Rule:

$$\Pr(L) = \prod_{i=1}^{n} \Pr(w_i | w_1^{i-1})$$

Goal: Estimate $P(w|h)$

View from Information Theory:

- L is an information source
- Goal: Find and exploit information in h about w

Measuring Model Quality

1. Perplexity [Jelinek *et al. 77*]

   Measures the average "surprise" of the model M upon seeing text T (cross-entropy):

$$\mathrm{PP}_M(T) = 2^{-\sum_{\mathbf{x}} P_T(\mathbf{x}) \cdot \log P_M(\mathbf{x})}$$

   - Meaningful only within same vocabulary and text
   - Does not consider acoustic confusability
   - Does not emphasize outliers

2. Acoustic Perplexity [deSouza]

3. Speech Decoder Entropy [Ferretti *et al. 89*]

4. Fraction of "surprises" [Bahl *et al. 89*]

Smoothing

- After seeing 38 million words of WSJ, new trigrams are still common:
  - 21% in 5,000 vocabulary
  - 31% in 20,000 vocabulary
- Rare linguistic events will always exist (but their impact may be diminished!)

Smoothing Methods:

- Assigning a “floor”
- Interpolating with the uniform distribution
- Good-Turing estimate [Good 53]
- Many other variants [Nadas 84, Church *et al. 91*, Ney *et al. 91*, Placeway*et al. 93*]

Assessing the Potential of Information Sources

1. Mutual Information (MI):

   Measures the reduction in uncertainty of Y when X is made known:

$$I(X:Y) \stackrel{\text{def}}{=} H(Y) - H(Y|X) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

   - Model may not be able to exploit the information
   - MI does not consider overlap with other sources
   - $I(X:Y) = 0$, $I(Z:Y) = 0$, yet $I(X,Z:Y) > 0$?

2. Training set perplexity [Huang *et al. 93*]

3. Shannon Game [Shannon 51, Mercer *et al. 92*]

Short-Term History (N-gram)

$$\Pr(w_i|h) = \Pr(w_i|w_{i-2}, w_{i-1})$$

Choosing N: tradeoff between detail and reliability

<u>Deficiencies:</u>

- "Blind" to any influence outside their scope
- Predictors defined by their ordinal position:

GOLD PRICES FELL TO
vs.
GOLD PRICES FELL YESTERDAY TO

Nonetheless, surprisingly powerful

Short-term Class History (Class N-gram)

$$P(w_i|h) = f\,(w_i|w_{i-1}, w_{i-2})$$

is replaced by one of:

$$\begin{aligned} P(w_i|h) &= f\,(w_i|w_{i-1}, g(w_{i-2})) \\ P(w_i|h) &= f\,(w_i|g(w_{i-1}), g(w_{i-2})) \\ P(w_i|h) &= f\,(g(w_i)|g(w_{i-1}), g(w_{i-2}))f\,(w_i|g(w_i)) \end{aligned}$$

1. Clustering by linguistic knowledge (POS)

2. Clustering by domain knowledge (ATIS)

3. Data Driven Clustering (NPOS)

   - Maximum Likelihood under HMMs [Jelinek 89]
   - Minimizing loss-of-MI [Brown *et al. 90*]
   - Soft clustering (“synonyms”) [Jelinek *et al. 90*]

Intermediate Distance

GOLD PRICES FELL TO
vs.
GOLD PRICES FELL YESTERDAY TO

Long-Distance bigrams: predict w_i based on w_{i-d}

d	1	2	3	4	5	6	7	8	9	10	1000
PP	83	119	124	135	139	138	138	139	139	139	141

- Data is unnecessarily fragmented

Long Distance (Triggers)

Basic element of long distance effect: $A \rightarrow B$

Define:

$$\begin{aligned} B &: \{w = B\} \\ A_\circ &: \{A \in h\} \end{aligned}$$

A natural measure for the utility of $A \rightarrow B$:

$$\begin{aligned} I(A_\circ : B) = \; & P(A_\circ, B) \log \frac{P(B|A_\circ)}{P(B)} \\ & + P(A_\circ, \overline{B}) \log \frac{P(\overline{B}|A_\circ)}{P(\overline{B})} \\ & + P(\overline{A_\circ}, B) \log \frac{P(B|\overline{A_\circ})}{P(B)} \\ & + P(\overline{A_\circ}, \overline{B}) \log \frac{P(\overline{B}|\overline{A_\circ})}{P(\overline{B})} \end{aligned}$$

Detailed Trigger Relations

- Binary triggers
- Count-Based triggers
- Distance-Based triggers
- . . .

$\Longrightarrow$ Look at the data!

Detailed Trigger Relations

- Binary triggers
- Count-Based triggers
- Distance-Based triggers
- . . .

$\Longrightarrow$ Look at the data!

<u>Conclusions:</u>

- Different behavior by different trigger pairs
- *Self-triggers* are powerful and robust
- Same-root triggers also strong
- Most of the potential is in high frequency pairs

Syntactic Constraints

- Number agreement
- Tense agreement
- . . .
- Grammaticality

Potential hard to assess, but is probably large

Combining Information Sources: Linear Interpolation

$$P_{\text{COMBINED}}(w|h) \stackrel{\text{def}}{=} \sum_{i=1}^{k} \lambda_i P_i(w|h)$$

$\lambda_i \geq 0$ and $\Sigma_i \lambda_i = 1$.

- λ_i's determined by EM (optimal wrt heldout data)
- Different λ_i's based on counts [Jelinek *et al. 80*]

<u>Advantages:</u>

- Extremely general
- Easy to implement and analyze
- Can't hurt

<u>Deficiencies:</u>

- Makes suboptimal use of components
- Generally inconsistent with components

Combining Information Sources: Backoff

1. Rank the various estimate
2. Use the first one that has enough information

The Backoff **N**-gram [Katz 87]:

$$P_n(w_n|w_1^{n-1}) = \begin{cases} (1-d)C(w_1^n) \;/\; C(w_1^{n-1}) & \text{if } C(w_1^n) > 0 \\ \alpha(C(w_1^{n-1})) \cdot P_{n-1}(w_n|w_2^{n-1}) & \text{if } C(w_1^n) = 0 \end{cases}$$

Deficiencies:

- Special case of linear interpolation, hence inferior
- Discontinuity around the backoff decision point
- Overestimation in N-gram
  (corrected in [Rosenfeld *et al. 92*])

Yet: simple, compact, powerful

The Maximum Entropy Principle

Seeking a single, combined estimate $P(\mathbf{x})$:

1. Reformulate every estimate as a *constraint* on $P(\mathbf{x})$.
2. Among P's satisfying all the constraints, choose the one with highest entropy (i.e. the "flattest").

Two Estimates for P(BANK||h)

The event space: $\{(h, w)\}$

h ends in "THE"	h ends in "OF"	. . .	. . .
. . . THE . . .	. . . OF . . .	.	.
. . . THE BANK	. . . OF . . .	.	.
. . . THE . . .	. . . OF BANK	.	.
. . . THE . . .	. . . OF . . .	.	.
. . . THE BANK	. . . OF . . .	.	.
. . . THE . . .	. . . OF . . .	.	.

$$\underset{h \text{ ENDS IN "THE"}}{\forall h} \quad P_{\text{BIGRAM}}(\text{BANK}|h) = 0.333$$

LOAN $\in h$	. . . LOAN . . . . . BANK . . . . . LOAN . . . . . BANK . . . LOAN . . . . .
LOAN $\notin h$	. . . . . BANK . . . . BANK . . . . .

$$\underset{\text{"LOAN"} \in \text{h}}{\forall h} \quad P_{\text{TRIGGER}}(\text{BANK}|h) = 0.020$$

A Joint Estimate for P(BANK||h)

	h ends in "THE"	h ends in "OF"	. . .
LOAN $\in h$	. . . . .	. . . . .	. . . . .
LOAN $\notin h$	. . . . .	. . . . .	. . . . .

$$\forall_{h \text{ ENDS IN "THE"}} h \quad P_{\text{BIGRAM}}(\text{BANK}|h) = 0.333$$

$$\Longrightarrow \quad E_{h \text{ ENDS IN "THE"}} [\ P_{\text{COMBINED}}(\text{BANK}|h)\ ] = 0.333$$

$$\forall_{\text{"LOAN"} \in \text{h}} h \quad P_{\text{TRIGGER}}(\text{BANK}|h) = 0.020$$

$$\Longrightarrow \quad E_{\text{"LOAN"} \in \text{h}} [\ P_{\text{COMBINED}}(\text{BANK}|h)\ ] = 0.020$$

Knowledge Sources as Constraints

<u>Generalization # 1:</u>

- Any subset S of the event space $\{(h, w)\}$
- Any desired expectation K

$$\underset{(h,w)\in S}{E} [\ P(h, w)\ ] = K$$

- Specify S by an *index function* f_S:

$$\sum_{(h,w)} [\ P(h, w) f_S(h, w)\ ] = K$$

<u>Generalization # 2:</u>

- Can use *any real-valued function* $f\ (h, w)$:

$$\langle f\ , P\rangle = K$$

- f is *a constraint function*
- $\langle f\ , P\rangle$ is the expectation of f under the desired P

$\Longrightarrow$ *Any* phonomenon can be modeled!

Maximum Entropy Formalism [Jaynes 57]

To derive a combined estimate $P(\mathbf{x})$:

1. Reformulate every estimate as a constraint on $P(\mathbf{x})$.
2. Among P's satisfying all the constraints, choose the one with highest entropy.

Given a partial estimate, define a *constraint function* $f_i(\mathbf{x})$, and constrain its expectation:

$$E_P f_i \stackrel{\text{def}}{=} \sum_{\mathbf{x}} P(\mathbf{x}) f_i(\mathbf{x}) = K_i$$

Then, a unique ME solution always exists:

$$P(\mathbf{x}) = \prod_i \mu_i^{f_i(\mathbf{x})}$$

"Generalized Iterative Scaling" [Darroch *et al. 72*] will converge to the correct μ_i's.

Generalized Iterative Scaling

<u>Input:</u>

- Constraint functions $f_i(\mathbf{x})$, desired values K_i.
- Training data: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$.

<u>Init:</u>

- Set $\mu_i^{(0)}$ to some arbitrary positive values.
- $P^{(0)}(\mathbf{x}) = \prod_i \mu_i^{(0)\, f_i(\mathbf{x})}$

<u>Iterate:</u>

- Compute $E_{P^{(k)}}[f_i]$ for all i.
- Update: $\mu_i^{(k+1)} = \mu_i^{(k)} \cdot \dfrac{K_i}{E_{P^{(k)}}[f_i]}$
- Set: $P^{(k+1)}(\mathbf{x}) = \prod_i \mu_i^{(k+1)\, f_i(\mathbf{x})}$

until $P^{(k)}$ converges.

Estimating Conditional Distributions

G.I.S. computes $P(\mathbf{x})$, but we need $P(w|h)$.

<u>Solutions:</u>

1. Estimate $P(h, w)$, then derive $P(w|h)$ [Lau *et al. 93*]
   Problem: modeling a space of size $O(V^{L+1})$

2. Modify ME constraints [Brown *et al.*]:

   Original: $$\sum_{(h,w)} P(h, w) \cdot f(h, w) = K$$

   Rewrite: $$\sum_{h} P(h) \cdot \sum_{w} P(w|h) \cdot f(h, w) = K$$

   Modify to: $$\sum_{h \in Tr} \sum_{w} P(w|h) \cdot f(h, w) = K$$

   - Now summing in space of size $\|\text{Training-Set}\| \cdot V$
   - ME solution = ML solution of exponential family.
     $\Longrightarrow$ can use gradient descent!

Minimum Discrimination Information (MDI)

- A prior $P_0(\mathbf{x})$
- A family $Q_\alpha(\mathbf{x})$ (intersection of constraints?)
- Goal: find $Q_0 \in Q_\alpha$ that 's "closest" to P_0

$$Q_0(\mathbf{x}) \stackrel{\text{def}}{=} \arg\min_\alpha D(Q_\alpha, P_0)$$

$$D(Q(\mathbf{x}), P(\mathbf{x})) \stackrel{\text{def}}{=} \sum_{\mathbf{x}} Q(\mathbf{x}) \log \frac{Q(\mathbf{x})}{P(\mathbf{x})}$$

- Q_0 is the MDI solution
- If P_0 is uniform, Q_0 is also the ME solution
- Used to adapt bigram to match new marginal constraints [DellaPietra *et al. 92*]

Assessing the ME approach

Advantages:

1. Intuitively appealing.
2. Extremely general:
   - Any knowledge source can be incorporated.
   - Constraints need not be independent/uncorrelated.
   - Constraints need not be derived from the data.
   - Constraints need not be consistent (with caveat).
3. Absorbes existing language models.
4. Algorithm is incrementally adaptive.
5. Guaranteed existence, uniqueness and convergence.
6. ME solution = ML solution of the exponential family:

$$P(\mathbf{x}) = \prod_i \mu_i^{f_i(\mathbf{x})}$$

Disadvantages:

1. Computationally expensive.
2. No bound on the convergence rate.
3. Convergence not guaranteed if inconsistent.

Formulating N-grams as ME Constraints

- The *constraint function* for $\{w_1, w_2\}$:

$$f_{\{w_1,w_2\}}(h, w) = \begin{cases} 1 & \text{if } h \text{ ends in } w_1, \text{ and } w = w_2 \\ 0 & \text{otherwise} \end{cases}$$

- The *desired* expectation:

$$K_{\{w_1,w_2\}} = \tilde{E}[f_{\{w_1,w_2\}}] \stackrel{\text{def}}{=} \frac{1}{N} \sum_{(h,w)\in Tr} f_{\{w_1,w_2\}}(h, w)$$

- The constraint on the *desired estimate* $P(w|h)$:

$$\sum_{h\in Tr} \sum_{w} P(w|h) \cdot f_{\{w_1,w_2\}}(h, w) = \tilde{E}[f_{\{w_1,w_2\}}]$$

Formulating Triggers as ME Constraints

- The *constraint function*:

$$f_{A \to B}(h, w) = \begin{cases} 1 & \text{if } A \in h,\ w = B \\ 0 & \text{otherwise} \end{cases}$$

- The *desired* expectation:

$$K_{A \to B} = \tilde{E}\,[f_{A \to B}] \stackrel{\text{def}}{=} \frac{1}{N} \sum_{(h,w) \in Tr} f_{A \to B}(h, w)$$

- The constraint on the *desired estimate* $P(w|h)$:

$$\sum_{h \in Tr} \sum_{w} P(w|h) \cdot f_{A \to B}(h, w) = \tilde{E}\,[f_{A \to B}]$$

Generalized Iterative Scaling Revisited

<u>Input:</u>

- Constraint functions $f_i(h, w)$, desired values K_i.
- Training data: $(h, w)_1, (h, w)_2, \ldots, (h, w)_n$.

<u>Init:</u>

- Set $\mu_i^{(0)}$ to some arbitrary positive values.
- $P^{(0)}(h, w)) = \prod_i \mu_i^{(0)\, f_i(h,w)}$

<u>Iterate:</u>

- For all i, compute $E_{P^{(k)}}[f_i] = \sum_{h \in Tr} \sum_w P(w|h) \cdot f_i(h, w)$
- For all i, update: $\mu_i^{(k+1)} = \mu_i^{(k)} \cdot \dfrac{K_i}{E_{P^{(k)}}[f_i]}$
- Set: $P^{(k+1)}(h, w) = \prod_i \mu_i^{(k+1)\, f_i(h,w)}$

until $P^{(k)}$ converges.

A Mini-Example: 2 Constraints

1. $$E_P[f_{\{\text{THE,BANK}\}}] = \sum_{h \in Tr} \sum_{w} P(w|h) \cdot f_{\text{THE,BANK}}(h, w)$$

$$= \sum_{\substack{h \in \text{TR} \\ h \text{ ends in THE}}} P(\text{BANK}|h)$$

2. $$E_P[f_{\text{LOAN}\rightarrow\text{BANK}}] = \sum_{h \in Tr} \sum_{w} P(w|h) \cdot f_{\text{LOAN}\rightarrow\text{BANK}}(h, w)$$

$$= \sum_{\substack{h \in \text{TR} \\ \text{LOAN} \in h}} P(\text{BANK}|h)$$

$\mu_{\{\text{THE,BANK}\}} = 1.1$ $\qquad$ $\mu_{\text{LOAN}\rightarrow\text{BANK}} = 1.2$

$$P(h, w) = \prod_{i=0}^{2} \mu_i^{f_i(h,w)} = \mu_0 \cdot 1.1^{f_{\{\text{THE,BANK}\}}(h,w)} \cdot 1.2^{f_{\text{LOAN}\rightarrow\text{BANK}}(h,w)}$$

	w_1	w_2	…	…	…	BANK	…	…	…	$w_{20,000}$
$P(h, w)$	μ_0	μ_0	…	…	…		…	…	…	μ_0
$P(w\|h)$	…	…	…	…	…		…	…	…	…

"USUALLY THE FIRST LOAN IS THE MOST ..."

Selecting Trigger Pairs

Consider:

> "The district attorney's office launched an investigation into loans made by several well connected banks."

- <u>Problem:</u> Will be used with trigram
- <u>Solution:</u> Modify the definition:

$$A_\circ \stackrel{\text{def}}{=} \{A \in w_1^{i-1}\}$$
$$\Downarrow$$
$$A_\circ \stackrel{\text{def}}{=} \{A \in w_1^{i-3}\}$$

- Call this measure MI-3g($A_\circ : B$)

Trigger Pairs

HARVEST ⇐ crop harvest corn soybean soybeans agriculture grain drought grains bushels

HARVESTING ⇐ crop harvest forests farmers harvesting timber trees logging acres forest

HASHEMI ⇐ iran iranian tehran iran's iranians lebanon ayatollah hostages khomeini israeli hostage shiite islamic iraq persian terrorism lebanese arms israel terrorist

HASTINGS ⇐ hastings impeachment acquitted judge trial district florida

HATE ⇐ hate my you her man me i love

HAVANA ⇐ cuban cuba castro havana fidel castro's cuba's cubans communist miami revolution

- Self triggers and same-root triggers are common.
- Heavy influence by frequency.

Combining *N*-grams and Triggers

vocabulary	top 20K words of WSJ corpus	
training set	5MW (WSJ)	
test set	325KW (WSJ)	
trigram perplexity (baseline)	173	173
ME experiment	top 3	top 6
ME constraints:		
unigrams	18400	18400
bigrams	240000	240000
trigrams	414000	414000
triggers	36000	65000
ME perplexity	134	130
% change	-23%	-25%

Class Triggers

$$\{\text{LOAN,LOAN'S}\} \Longrightarrow \{\text{BANK,BANKER}\}$$

$$f_{AA \to BB}(h, w) = \begin{cases} 1 & \text{if } (\exists A, A \in AA, A \in h) \wedge w \in BB \\ 0 & \text{otherwise} \end{cases}$$

vocabulary	top 20,000 words of WSJ corpus	
training set	300KW (WSJ)	
test set	325KW (WSJ)	
model	word triggers	class triggers
ME constraints:		
unigrams	9017	9017
word self-triggers	2658	—
baseform self-triggers	—	2409
training-set perplexity	745	740
test-set perplexity	888	870

Long-Distance N-gram Constraints

- The *constraint function*:

$$f^{[j]}_{\{w_1,w_2\}}(h,w) = \begin{cases} 1 & \text{if } h = w_1^{i-1}, w_{i-j} = w_1 \text{ and } w = w_2 \\ 0 & \text{otherwise} \end{cases}$$

- The *desired* expectation:

$$K^{[j]}_{\{w_1,w_2\}} = \tilde{E}\,[f^{[j]}_{\{w_1,w_2\}}] \stackrel{\text{def}}{=} \frac{1}{N} \sum_{(h,w)\in \text{TRAINING}} f^{[j]}_{\{w_1,w_2\}}(h,w).$$

- The constraint on the *desired estimate* $P(w|h)$:

$$\sum_h \tilde{P}(h) \sum_w P(w|h) \cdot f^{[j]}_{\{w_1,w_2\}}(h,w) \;=\; \tilde{E}\,[f^{[j]}_{\{w_1,w_2\}}]$$

A Large Maximum Entropy Model

vocabulary	top 20K words of WSJ corpus		
test set	325KW		
training set	1MW	5MW	38MW*
trigram perplexity (baseline)	269	173	105
ME constraints:			
unigrams	13130	18421	19771
bigrams	65397	240256	327055
trigrams	79571	411646	427560
distance-2 bigrams	67186	280962	651418
distance-2 trigrams	65600	363095	794818
word triggers (top 3)	20209	35052	43066
training time (alpha-days)	< 0.5	12	~ 200
test-set perplexity	203	123	86
% change	-24%	-29%	-18%

* 38MW model:

- Used high cutoffs
- *One third* the parameters of the baseline trigram!

Effect of Individual Knowledge Sources

vocabulary	20,000 words	
training set	1MW	
test set	325KW	
	perplexity	%change
trigram (baseline)	269	—
ME models:		
dist.-1 *N*-grams + dist.-2 *N*-grams	249	-8%
dist.-1 *N*-grams + word triggers	208	-23%
dist.-1 *N*-grams+dist.-2 *N*-grams+trigs	203	-24%

Computational Issues

- Bottleneck: collecting expectations $\rightarrow$ parallelizable
- Memory needs: $>$64MB ROM, $>$20MB writable

Solutions (not all implemented):

1. Paralellization:
   - Multi-processor machines $\Rightarrow$ not enough memory
   - SIMD $\Rightarrow$ not enough private memory

   $\Longrightarrow$ Conventional workstations
2. Reducing computation per iteration:
   - Reduce constraints (e.g. cutoffs)
   - Estimate expectations by sampling
3. Speeding up convergence:
   - Limit maximum number of constraints per word
   - Dynamically modify step size
   - Use alternative search methods

Not a problem for other ME tasks [Ratnaparkhi$^+$ 94]

Adaptation

- Adaptation $\approx$ long-distance modeling

3 Paradigms of Adaptation:

1. **Within-domain adaptation**
   - Language is mixture of "sublanguages"
   - Goal is to track sublanguage mixture & shift
   - Use triggers!
2. **Cross-domain adaptation**
   - Training domain $\neq$ test domain
   - High OOV rate
   - High rate of new bigrams & trigams
   - Use caches!
3. **Limited-data domain adaptation**
   - Lots of general language data
   - Moderate amounts of data from test domain
   - Most likely in real world

Caches

Selective Unigram Cache:

- Observation: strong self-correlation ("self-trigger").
- *But:* phenomenon depends on frequency of the word:

$$\text{BRAZIL} \rightarrow \text{BRAZIL} \quad \text{vs.} \quad \text{THE} \rightarrow \text{THE}$$

- Optimally, use bayesian analysis to capture effect.
- Approximation: cache only "rare" words.

Conditional Bigram Cache:

- Bigram cache can be viewed as set of unigram caches
- $\Longrightarrow$ should only contribute if last word is a unigram cache "hit".
- Alternate view: context-dependent interpolation weights.

Within-Domain Adaptation

vocabulary	top 20K words of WSJ corpus		
training set	1MW	5MW	38MW
test set	325KW	325KW	325KW
trigram perplexity (baseline)	269	173	105
Maximum Entropy:			
test-set perplexity	203	123	86
% change	-24%	-29%	-18%
ME + trigram + caches:			
test-set perplexity	163	108	71
% change	-39%	-38%	-32%

- Less training data $\Longrightarrow$ more cache impact
- Adaptive-LM(5MW) $\approx$ trigram(38MW)

Contribution of Individual Components

vocabulary	top 20K words of WSJ corpus		
training set	1MW	5MW	38MW
test set	325KW	325KW	325KW
trigram perplexity (baseline)	269	173	105
trigram + caches:			
test-set perplexity	193	133	88
% change	-28%	-23%	-17%
Maximum Entropy:			
test-set perplexity	203	123	86
% change	-24%	-29%	-18%
ME + trigram:			
test-set perplexity	191	118	75
% change	-29%	-32%	-28%
ME + trigram + caches:			
test-set perplexity	163	108	71
% change	-39%	-38%	-32%

Cross-Domain Adaptation

The Need for Cross-Domain Adaptation:

vocabulary	top 20,000 words of WSJ corpus	
training set	WSJ (38MW)	
test set	WSJ (325KW)	AP (420KW)
OOV rate	2.2%	3.9%
trigram hit rate	60%	50%
trigram perplexity	105	206

Results:

vocabulary	top 20K words of WSJ corpus
training data	38MW (WSJ)
test data	420KW (AP)
trigram (baseline)	
perplexity	206
Maximum Entropy	
perplexity	170
% change	-17%
ME + trigram + caches	
perplexity	130
% change	-37%

Limited-Data Domain Adaptation

vocabulary	top 20K words of WSJ
trigger derivation data	38MW (WSJ)
training data	5MW (AP)
test data	420KW (AP)
trigram (baseline) perplexity	 170
Maximum Entropy perplexity % change	 135 -21%
interpolated model perplexity % change	 114 -33%

LM Interface to SPHINX-II

Currently 3 passes in SPHINX-II

- forward+backward:
  - use a (static) bigram - no adaptation here!
  - produce a lattice

- A*:
  - uses the lattice
  - produces N-best hypotheses
  - has access to entire history $\Longrightarrow$ can use any LM

General LM interface to the A* pass:

- given: partial document h (a path in the lattice)
- returns: array of probabilities
- backtracking of ME model $\Longrightarrow$ CPU intensive

Within-Domain Error Rate Reduction

- 4 speakers, 20 documents, 424 sentences
- Some language drift

training data	38MW (WSJ 87-89)		
test data	424 sentences (WSJ 90)		
language model	W.E.R.	%chng	source of information
trigram	18.4	—	last two words
adaptive:			
isolated sents	17.3	-6%	same sentence
unsupervised	16.7	-9%	+ prev. sents(unsupervsd)
supervised	16.1	-13%	+ prev. sents(supervised)

Cross-Domain Error Rate Reduction

- Same configuration as before, only different data

training data	38MW (WSJ)	
test data	206 sentences (AP)	
language model	word error rate	%change
trigram (baseline)	22.1%	—
supervised adaptation	19.8%	-10%

Summary of Thesis Work

- Introduced Maximum Entropy to Language Modeling.
- First and only successful integration of short- and long-term effects.
- Improved adaptation by better caching techniques.

Practical Significance of Thesis:

- Reduced perplexity by 32%–39%
- Reduced error rate by 10%–14%
- Created machinery for consistent integration of disparate knowledge sources.

Related Contributions

SLM Toolkit, available by anonymous ftp:

- Accelerates research start-up time
- Modular
- Extensible

Currently used by (partial list):

- Both CMU speech groups
- CMT
- IBM R&D, Maryland
- Microsoft Corp.
- University of Karlsrue
- ARPA Speech community

Future Directions

Fundamental Issues Revisited

1. Identifying knowledge sources
2. Combining knowledge sources
3. Adaptation (domains, styles, time shift)
4. Exploiting the deep structure of language.

Research Agenda

- Adaptation: unlimited, smart vocabulary
- Adaptation: dynamic reintegration of knowledge sources
- Adaptation: separating content from style
- Improving the predictors [Lafferty *et al.* 92]
- Combining acoustic and linguistic evidence

Unlimited, Smart Vocabulary

The challenge: Reduce Out-Of-Vocabulary rate, given lots of general data, and possibly a bit of specific data.

Types of OOVs:

1. data-dependent OOVs (insufficient training data)
2. within-domain OOVs (new topics, time shift)
3. cross-domain OOVs

Answer questions like:

- OOV rate as function of types & amt. of training data
- how much new domain data is needed?
- what size vocabulary to start with?

Cache Closures:

1. Morphologically derived closures
2. Topic-related closures
3. Data-driven closures