

**Theory and Practice
of Acoustic Confusability**

Harry Printz Peder Olsen

**IBM
Watson Research Center**

Plan of Talk

- Motivation
- Perplexity and Maximum Likelihood
- Acoustic Perplexity: Intuition and Definition
- How to Compute Confusability
- Typical Results
- How to Train Language Models
- Decoding Results
- Summary

Motivation

Philip Clarkson, Tony Robinson, “The Applicability of Adaptive Language Modelling for the Broadcast News Task,” *Intl Conference on Spoken Language Processing*, 1999

This paper has described two simple adaptive language models, and shown that while they lead to substantial reductions in perplexity over the baseline Broadcast News language model, they do not result in improved recognition performance. . . . [T]hese results, as well as those given in several other papers, show that even fairly large reductions in perplexity are no guarantee of a reduction in word error rate.

Perplexity and Maximum Likelihood

- seek to predict a *future* x_i from a *history* h_i
- λ is a large vector, defining $p_\lambda(x | h)$
- $P_\lambda(\mathcal{C}) = \prod_{i \in \mathcal{C}} p_\lambda(x_i | h_i)$, trained via $\operatorname{argmax}_\lambda P_\lambda(\mathcal{C})$

For corpus \mathcal{C} of size N , and model p_λ , define *lexical perplexity*

$$Y_L(p_\lambda, \mathcal{C}) = \exp \left\{ -\frac{1}{N} \log P_\lambda(\mathcal{C}) \right\}$$

$P_\lambda(\mathcal{C}) \uparrow$ iff $Y_L(p_\lambda, \mathcal{C}) \downarrow$

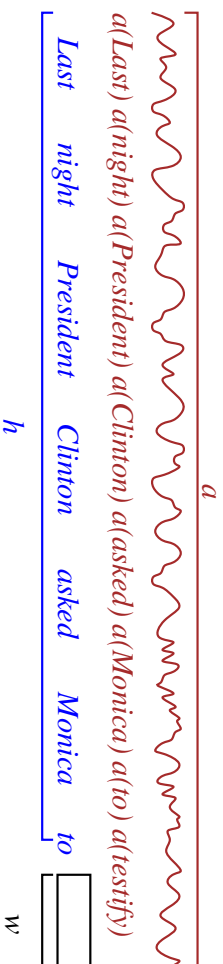
Acoustic Perplexity: Intuition

Text creation



- prediction via $\operatorname{argmax}_w p(w | h)$
- training via $\operatorname{argmax}_\lambda P_\lambda(\mathcal{C})$

Speech recognition



- prediction via $\operatorname{argmax}_w p(w | a h)$
- training via $\operatorname{argmax}_\lambda P_\lambda(\mathcal{C} | \mathcal{A})$

Acoustic Perplexity: Definition

Given p_λ , \mathcal{C} (text) and \mathcal{A} (acoustics of \mathcal{C}), minimize

$$Y_A(p_\lambda, \mathcal{C}, \mathcal{A}) = \exp \left\{ -\frac{1}{N} \log P_\lambda(\mathcal{C} | \mathcal{A}) \right\}$$

Y_A is the *acoustic perplexity*.

We compute acoustic perplexity just as we do textual perplexity.

$$P(\mathcal{C}) = \prod_{i \in \mathcal{C}} p(w_i | h_i)$$

w_i	word at i
h_i	history at i

$$P(\mathcal{C} | \mathcal{A}) = \prod_{i \in \mathcal{C}} p(w_i | a(h_i w_i r_i) h_i)$$

w_i	word at i
h_i	history at i
r_i	right ctx at i
$a(h_i w_i r_i)$	acoustics \mathcal{A}

Observe

$$p(w_i | a(h_i w_i r_i) h_i) = \frac{p(a(h_i w_i r_i) | h_i w_i) p(w_i | h_i)}{\sum_{x \in V} p(a(h_i w_i r_i) | h_i x) p(x | h_i)}$$

We introduce

$$p(a(h w r) | h x) \quad \text{the acoustic encoding prob} \\ \text{of } x \text{ as } w \text{ in context } h r$$

Unfortunately

- we don't know how to compute $p(a(h w r) | h x)$
- we have woefully inadequate joint corpora \mathcal{C} , \mathcal{A}

Both problems are solved in this talk.

How to Compute Confusability

Computational simplification

$$p(a(h\ w\ r) \mid h\ x) \approx p(a(h) \mid h) p(a(w) \mid h\ x) p(a(r))$$

Express x by its pronunciation(s) $\{l(x)\}$

$$p(a(w) \mid h\ x) = \sum_{l(x) \in x} p(a(w) \mid h\ x\ l(x)) p(l(x) \mid h\ x)$$

$$p(a(w) \mid h\ x\ l(x)) = p(a(w) \mid h\ l(x))$$

Use a model $\hat{a}(w)$ of true acoustics $a(w)$

$$p(a(w) \mid h\ l(x)) \approx p(\hat{a}(w) \mid h\ l(x))$$

Consider phones $a \in \hat{a}(w)$ and $g \in l(x)$; ignore context h

$$p(a \mid g\ h) \approx p(a \mid g)$$

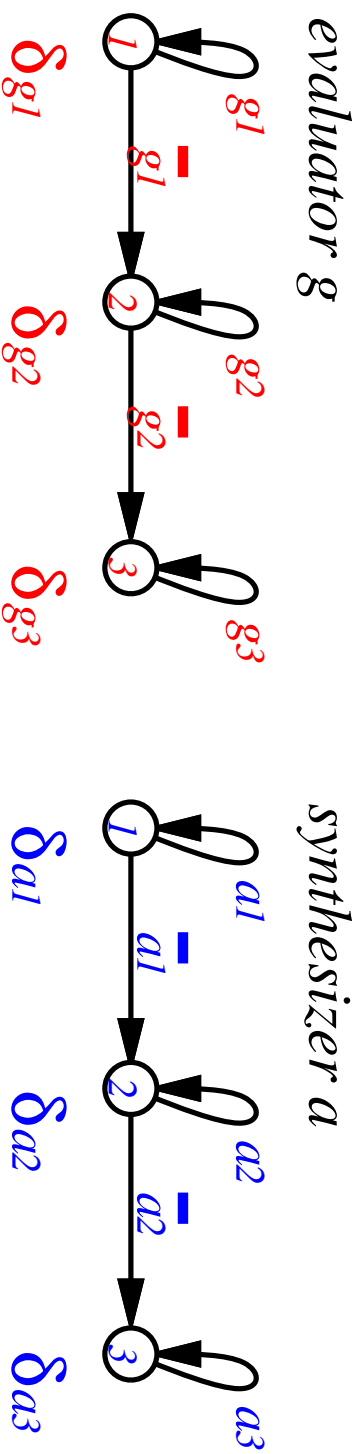
Method extends easily to full words, in context

How to Compute Confusability

Given models g (evaluator) and a (synthesizer); we seek

$$p(a \mid g) \quad \text{likelihood of } a\text{'s observations, evaluated by } g$$

g and a modeled by three-state, continuous-density models



evaluation by g of observation sequence

- *path* from g 's state 1 to state 3, say 122233, $p = \bar{g}_1 g_2 g_2 \bar{g}_2 g_3$
- *likelihood sequence* from densities $\delta_{g_1} \delta_{g_2} \delta_{g_2} \delta_{g_2} \delta_{g_3}$

synthesis by a of observation sequence

- *path* from a 's state 1 to state 3, say 112223, $p = a_1 \bar{a}_1 a_2 a_2 \bar{a}_2$
- *data sequence* from densities $\delta_{a_1} \delta_{a_2} \delta_{a_2} \delta_{a_2} \delta_{a_2}$

Let $\kappa(\delta_a | \delta_g)$ be δ_g 's observation likelihood for data from δ_a .

Then for joint path $\pi = \langle 1_g \ 1_a \rangle \langle 2_g \ 1_a \rangle \langle 2_g \ 2_a \rangle \langle 2_g \ 2_a \rangle \langle 3_g \ 2_a \rangle \langle 3_g \ 3_a \rangle$

$$p(a \mid g \pi) = \kappa(\delta_{a_1} | \delta_{g_1}) \kappa(\delta_{a_1} | \delta_{g_2}) \kappa(\delta_{a_2} | \delta_{g_2}) \kappa(\delta_{a_2} | \delta_{g_2}) \kappa(\delta_{a_2} | \delta_{g_2}) \kappa(\delta_{a_2} | \delta_{g_3})$$

$$p(\pi) = \bar{g}_1 a_1 g_2 \bar{a}_1 g_2 a_2 \bar{g}_2 a_2 g_3 \bar{a}_2$$

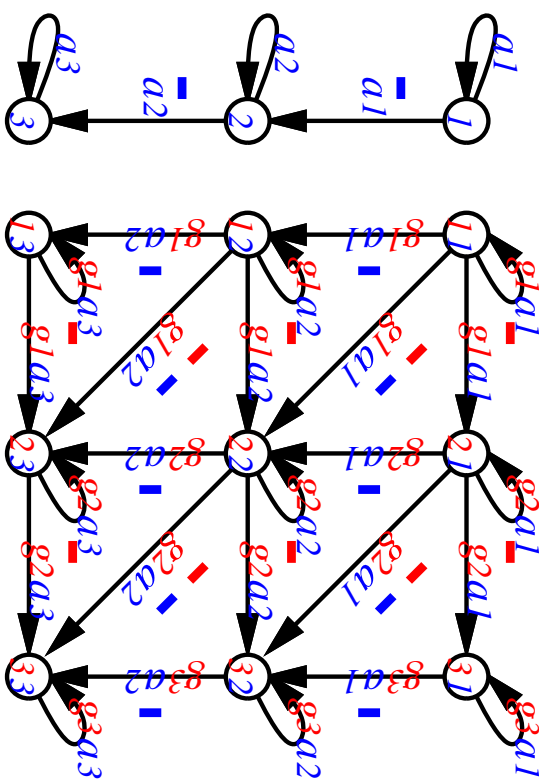
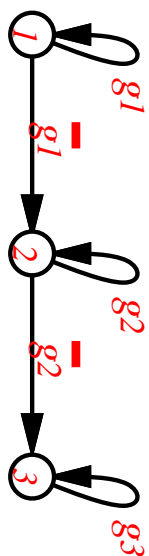
$$p(a \pi \mid g) = p(a \mid g \pi) p(\pi)$$

How to Compute Confusability

- have $p(a \pi | g)$ for one path π of length 5
- want $p(a | g) = \sum_{\pi \in \Pi} p(a \pi | g)$ for all paths, all lengths!

We compute this now. Strategy:

- form $g \times a$
- express path probability as likelihood flow within this machine
- determine $n + 1$ -path flow by dp over n -path flows
- represent likelihood flow by a linear transform
- iterate and sum this transform to accumulate over all paths



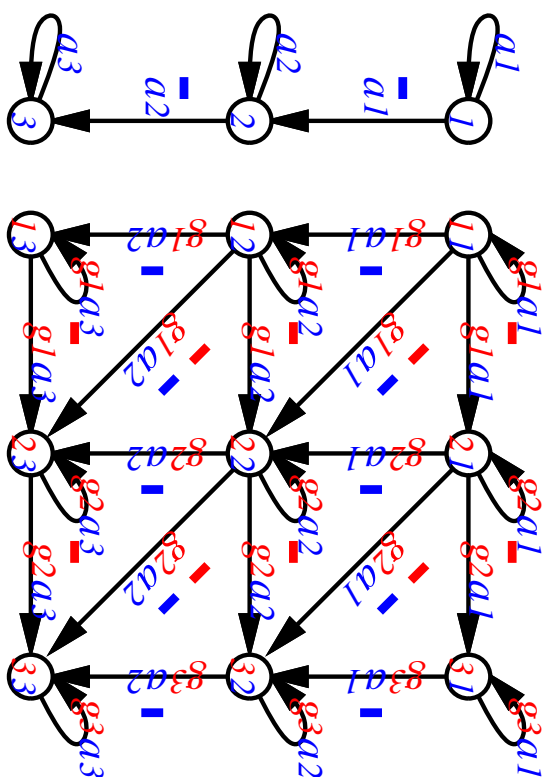
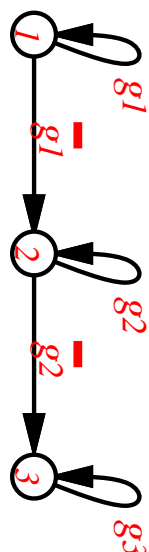
For phones g and a , machine $g \times a$ has 9 states.

Consider states s and d of $g \times a$, then

$$M_{ds} = \begin{cases} \tau(s \rightarrow d) \cdot \kappa(s \rightarrow d) & \text{if } s \rightarrow d \text{ in } g \times a \\ 0 & \text{otherwise} \end{cases}$$

τ is the $s \rightarrow d$ transition probability, κ is the a | g -likelihood

M has 9×9 elements, only 25 of them non-zero



{ length 5 paths $\langle g_1 a_1 \rangle \rightarrow \langle g_3 a_3 \rangle$ } = { one-step extensions of length 4 paths $\langle g_1 a_1 \rangle \rightarrow \langle g_2 a_2 \rangle, \langle g_2 a_2 \rangle$ or $\langle g_3 a_3 \rangle$ }

$$v_5^{\langle g_3 a_3 \rangle} = v_4^{\langle g_2 a_3 \rangle} \cdot \bar{g}_2 a_3 \cdot \kappa(\delta_{a_3} | \delta_{g_2}) + v_4^{\langle g_2 a_2 \rangle} \cdot \bar{g}_2 \bar{a}_2 \cdot \kappa(\delta_{a_2} | \delta_{g_2}) + v_4^{\langle g_3 a_2 \rangle} \cdot g_3 \bar{a}_2 \cdot \kappa(\delta_{a_2} | \delta_{g_3}) + v_4^{\langle g_3 a_3 \rangle} \cdot g_3 a_3 \cdot \kappa(\delta_{a_3} | \delta_{g_3})$$

where $v_t^{\langle g_i a_j \rangle}$ is the likelihood mass in state $\langle g_i a_j \rangle$ at time t .

$$v_5 = M v_4$$

How to Compute Confusability

$$p(a | g) = v_0^{\langle g_3 \ a_3 \rangle} + v_1^{\langle g_3 \ a_3 \rangle} + v_2^{\langle g_3 \ a_3 \rangle} + \dots$$

the sum over all length 0, length 1, length 2, ... paths.

$$v_0^T = [\underbrace{1}_{\langle g_1 \ a_1 \rangle \text{ elt}} \quad 0 \quad \dots \quad \underbrace{0}_{\langle g_3 \ a_3 \rangle \text{ elt}}]$$

$$v_0 = I v_0 \quad v_1 = M v_0 \quad v_2 = M v_1 = M^2 v_0 \quad \dots$$

We now sum over all paths of length 0, length 1, length 2, ...

$$v_\infty = v_0 + v_1 + v_2 + \dots = (I + M + M^2 + \dots)v_0 = (I - M)^{-1}v_0.$$

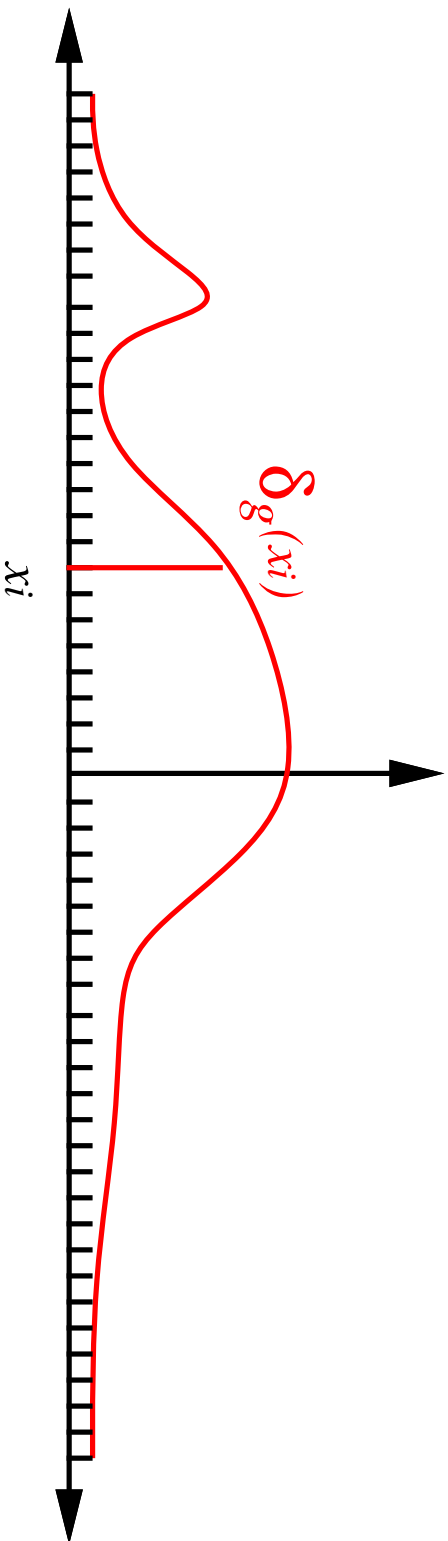
Only one matrix element needed!

$$\xi(a | g) = v_\infty^{\langle g_3 \ a_3 \rangle} = [(I - M)^{-1}v_0]^{\langle g_3 \ a_3 \rangle} = [(I - M)^{-1}]_{g_1}$$

Definition of $\kappa(\delta_a \mid \delta_g)$

δ_g , δ_a respectively evaluator and synthesizer leaf densities

One-dimensional space Ω , sampled D times, interval Δx .



Synthesize N observation data points in Ω according to δ_a

- $\sim N\delta_a(x_1)\Delta x$ fall into $[x_1, x_1 + \Delta x)$
- $\sim N\delta_a(x_2)\Delta x$ fall into $[x_2, x_2 + \Delta x)$
- \dots

Total likelihood according to δ_g of observing all N points

$$\Lambda(\delta_a \mid \delta_g) = \delta_g(x_1)^{N\delta_a(x_1)\Delta x} \delta_g(x_2)^{N\delta_a(x_2)\Delta x} \dots \delta_g(x_D)^{N\delta_a(x_D)\Delta x}$$

Expected per-point log-likelihood

$$\frac{1}{N} \log \Lambda(\delta_a \mid \delta_g) = \sum_i \delta_a(x_i) \log \delta_g(x_i) \Delta x$$

Taking the limit as $\Omega \rightarrow [-\infty, +\infty]$, $D \rightarrow \infty$

$$\frac{1}{N} \log \Lambda(\delta_a \mid \delta_g) = \int \delta_a(x) \log \delta_g(x) dx$$

Define

$$\log \kappa(\delta_a \mid \delta_g) = \int \delta_a(x) \log \delta_g(x) dx$$

Getting a Probability

$$p(l(w) | l(x)) = \frac{\xi(l(w) | l(x))^{1+\lambda}}{\sum_{l(w)' \in B} \xi(l(w)' | l(x))^{1+\lambda}}$$

Normalization required due to tiny ξ magnitudes.

λ required due to distribution sharpness.

Experimentally, $\lambda = -0.86$ works well.

$l(x) =$	B A O S T A X N	$\log_{10} p_\lambda(l(w) l(x))$
w	$l(w)$	$\lambda = 0.00$ $\lambda = -0.86$
<i>Boston</i>	B A O S T A X N	-0.00 -0.57
<i>Austin</i>	A O S T A X N	-5.92 -1.40
<i>Baden</i>	B A O D X A X N	-10.59 -2.05
<i>busted</i>	B A H S T I X D D	-10.73 -2.07
<i>bossed</i>	B A O S T D	-11.54 -2.19
<hr/>		
$l(x) =$	<i>Dallas</i>	$\log_{10} p_\lambda(l(w) l(x))$
$l(w)$	pronunciation	$\lambda = 0.00$ $\lambda = -0.86$
<i>Dallas</i>	D A E L A X S	-0.00 -0.98
<i>Dulles</i>	D A H L A X S	-6.37 -1.87
<i>Della</i>	D E H L A X	-7.09 -1.97
<i>ballots</i>	B A E L A X T S	-8.85 -2.22
<i>gala</i>	G A E L A X	-8.91 -2.23

Predicting Language Model Performance

Intuition and example lead

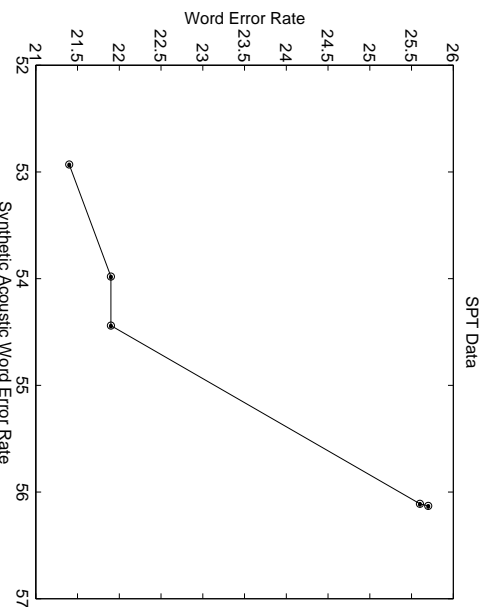
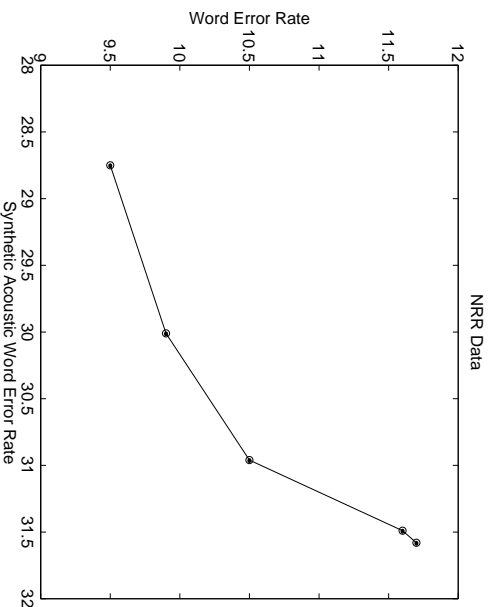
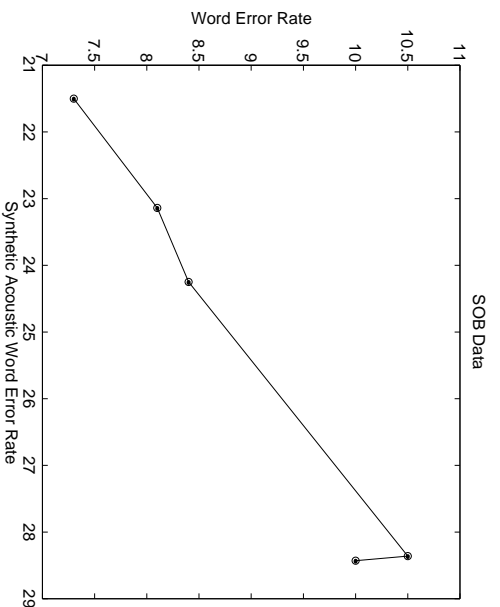
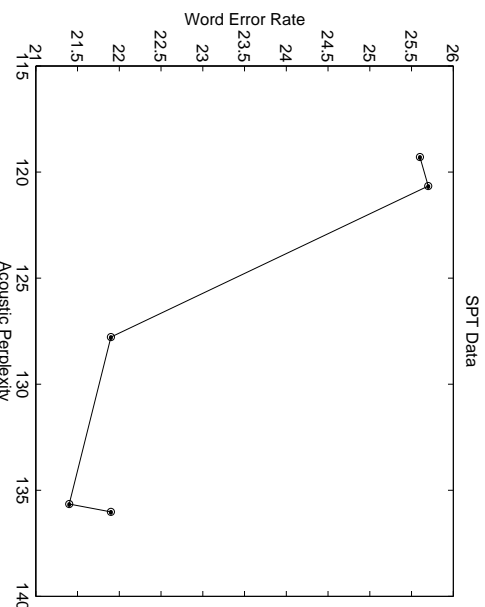
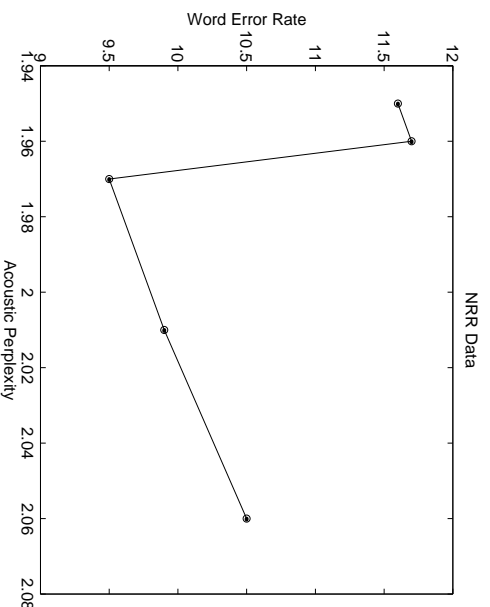
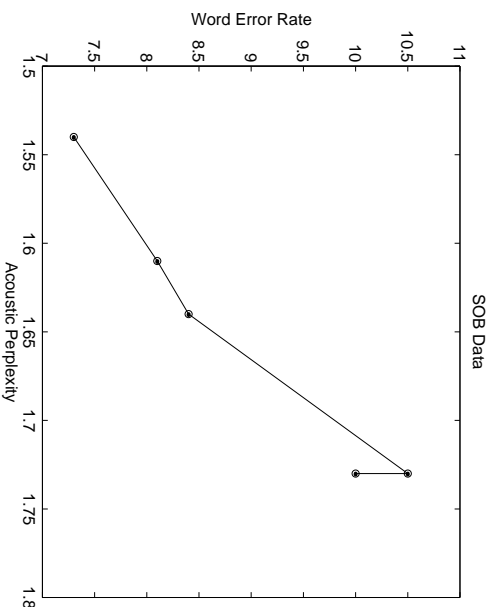
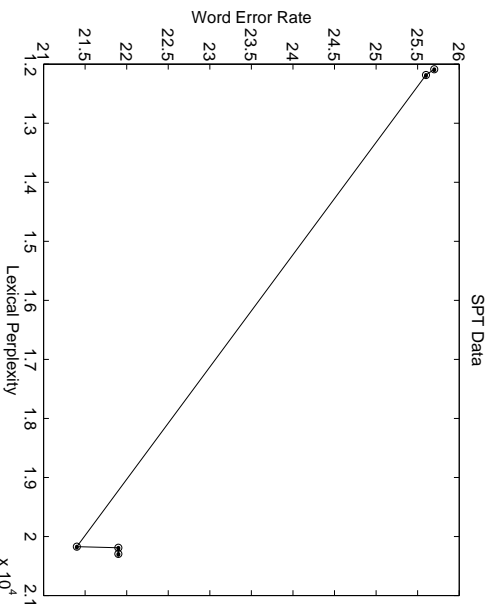
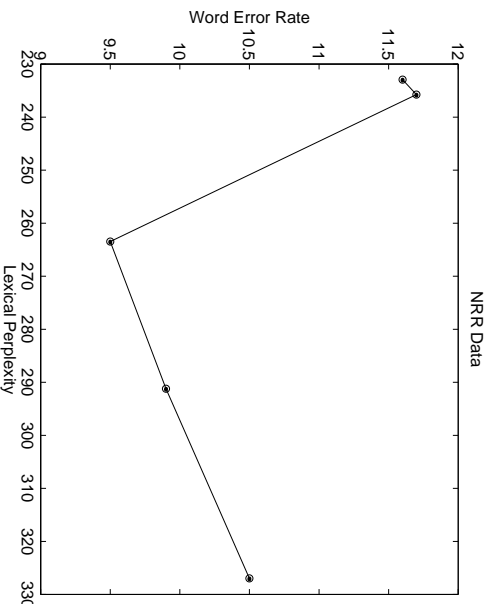
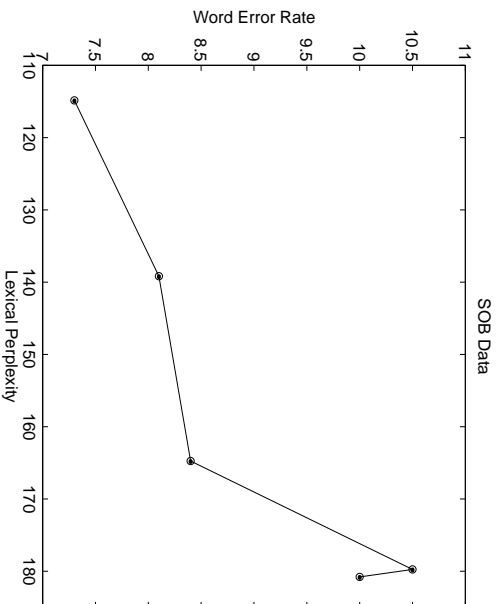
from $P(\mathcal{C})$ *lexical perplexity*
to $P(\mathcal{C} | \mathcal{A})$ *acoustic perplexity*

But what is the appropriate predictor of language model performance?

$$\begin{array}{ccccc} & x & & y & & z \\ p(x | a(x) h) & & p(y | a(y) h) & & p(z | a(z) h) \\ 1 - p(x | a(x) h) & & 1 - p(y | a(y) h) & & 1 - p(z | a(z) h) \end{array}$$

Consider the *synthetic acoustic word error rate* (SAWER)

$$S_A(P, \mathcal{C}) = E[\text{WER}] = 100 \cdot \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} (1 - p(w_i | a(w_i) h_i))$$



Training Language Models

We seek $p(w | h)$ minimizing

$$S_A(P, \mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} (1 - p(w_i | a(w_i) h_i))$$

with

$$p(w|a(w), h) = \frac{p(a(w)|w h)p(w|h)}{\sum_{x \in V} p(a(w)|x h)p(x|h)}.$$

Equivalently, for each h we separately *maximize*

$$s_h(\lambda) = \sum_{w \in V} c(w, h) \left(\frac{p(a(w)|w h)\lambda_{wh}}{\sum_{x \in V} p(a(w)|x h)\lambda_{xh}} \right)$$

where $\lambda_{wh} = p(w | h)$.

Constrained Gradient Method

Theorem 1 (Constrained Gradient Method) Take $s(\lambda)$ as defined above. Consider $\lambda = (\lambda_1, \dots, \lambda_m)$ satisfying $\sum_{i=1}^m \lambda_i = 1$, with $\lambda_i \geq 0$. Let $\hat{\lambda} = (\hat{\lambda}_1, \dots, \hat{\lambda}_m)$ where

$$\hat{\lambda}_i = \lambda_i + \epsilon \lambda_i \frac{\partial s}{\partial \lambda_i}$$

Then there exists $\epsilon > 0$ such that

$$\begin{aligned} s(\hat{\lambda}) &\geq s(\lambda) \\ \sum_{i=1}^m \hat{\lambda}_i &= 1 \\ \hat{\lambda}_i &\geq 0 \end{aligned} \tag{1}$$

where equality in (1) is obtained only if $\lambda_i(\partial s / \partial \lambda_i) = 0$ for all i .

Experimental Results

p_0	uniform model
p_1	raw unigram model
s_0	iteratively trained from p_0 on uniform corpus
s_1	iteratively trained from p_1 on unigram corpus
\bar{p}	$0.1s_0 + 0.1p_0 + 0.4s_1 + 0.4p_1$

	SOB	NRR	SPT	ALL
p_0	78.8%	76.2%	87.4%	84.9%
s_0	85.0%	84.4%	90.1%	88.7%
p_1	40.9%	47.3%	59.7%	55.5%
s_1	51.1%	58.3%	66.1%	63.0%
\bar{p}	40.5%	44.8%	59.8%	55.2%

Applications and Extensions

Language model compression

$b(w | h)$ bigram lm, xyz a trigram, count $c(xyz)$

$$\Delta_{xyz} = \frac{1}{N}(S_A(P_b, \mathcal{C}) - S_A(P_{b+xyz}, \mathcal{C}))$$

Maxent feature selection

Recognizer vocabulary selection

$$\Delta_u = \frac{1}{N}(S_A(P_V, \mathcal{C}) - S_A(P_{V \cup \{u\}}, \mathcal{C}))$$

Channel-adapted (translation) language models

Summary

- reviewed perplexity and maximum likelihood
- introduced and defined acoustic perplexity and SAWER
- demonstrated their properties and advantages
- derived an efficient algorithm for computing confusability
- presented experimental evidence backing this approach
- presented a language model training procedure and results
- described applications and extensions

$l(w)$	$\log_{10} p(l(w) \mid \textit{nostril})$
nostril	-0.230
austral	-1.384
nostrils	-1.632
mistral	-1.650
astral	-1.737

$l(w)$	$\log_{10} p(l(w) \mid \textit{ardor})$
ardor	-1.012
order	-2.014
otter	-2.149
eider	-2.281
harder	-2.287